
ATARI
ST COMPUTER
Die Fachzeitschrift für den ATARI-ST Anwender.
Juli ´88
DM 7,-
Ös. 56,- Sfr. 7,-
7
PUZZLESPRACHE
-Modula-2-Systeme im Überblick
CAMPUS DRAFT
-Der kleine CAD-Bruder
"GABI" SUCHT ANSCHLUSS
-Schreibmaschine & ST
DISKMASTER
-Formate selbst gestrickt
MODULA 2







## Wo der Anblick ein Überblick ist.

Auf einen Blick können wir die Software, die wir vertreiben, nur anschnittsweise vorstellen und wählen an dieser Stelle den kurzen Weg der Auflistung. Ausführliche Informationen zu einzelnen Produkten fordern Sie bitte schriftlich bei uns an.

Wir haben drei Programmiersprachen veröffentlicht, *Megamax C* (dazu die *Editor Toolbox*), *Megamax Modula-2* und schließlich *Imagic*, die Sprache, die Bilder schreibt.

*Signum!Zwei* ist sicher eine der umfassendsten und beliebtesten Textverarbeitungen, die zur Zeit zu haben sind, weil sie vielfältigen Anwenderbedürfnissen gerecht wird. Sie verfügt unter anderem über Mehrspaltensatz, freie Formeldefinition und Fremdsprachenzeichen. Sie hat einen eigenen Fonteditor und kann Grafiken von *STAD*, unserem professionellen Zeichenprogramm, einbinden. Dazu gibt es mittlerweile über 300 verschiedene Zeichensätze sowie eine Reihe nützlicher Zusatzprogramme.

Böse Zungen sprechen oft von der Unzulänglichkeit von Computern bei der Korrespondenz. *Daily Mail* wurde entwickelt, um dieser langgehegten Ansicht entgegenzutreten. Das Programm erledigt komfortabel Ihre tägliche Post, speichert Briefe und Adressen und kann Texte zur Weiterverarbeitung ausgeben.

## Wo der Teufel im Detail steckt.

Für creative User haben wir vor einiger Zeit eine Serie von Hilfsprogrammen vorgestellt, die wir unter dem Decknamen *Utility Series* anbieten. Bisher sind zwei Programme erschienen, weitere werden folgen. Beide Programme sind in einer neuen Version zu haben:

*FlexDisk*, die Ramdisk für den blitzschnellen Zugriff auf alle Daten. Die Größe läßt sich dabei beliebig einstellen, – ganz nach Bedarf. Accessories und Autoordner können jetzt von einem beliebigen Laufwerk gestartet werden, damit ist immer das nötige Werkzeug zur Verfügung. Die neue *Version 1.2.* ist nun noch schneller als die vorherigen Versionen.

Mit dem *Harddisk Utility* macht man Sicherheitskopien von der Harddisk und zwar ohne Umstände. Bei beliebiger Dateigröße sichert man auch sehr große Datenbestände auf Diskette. Durch Rückspeichern auf eine beliebige Harddisk oder gar Ramdisk können Dateien mit diesem Programm von einem Massenspeicher auf einen anderen transferiert werden. Das Harddisk-Backup Programm ist einfach, sicher und schnell zu bedienen und natürlich GEM eingebunden. Die neue *Version 2.0* ist beim Rückspeichern mehr als drei mal so schnell und hat eine Reihe neuer Features.

Registrierte Anwender erhalten die aktuellen Versionen komplett mit jeweils neuer Anleitung (40 bzw. 16 Seiten) gegen Einsendung der alten Originaldiskette und jeweils 20,– DM.

## Aktuell? Aktuell.

Zur CeBIT in Hannover ist sie zum ersten Mal erschienen: *die soft.* 20 Seiten im Großformat 25 x 35 cm informieren alle Interessierten über unsere Produkte und das, was wir sonst so tun. *die soft.* soll ab jetzt etwa 4 mal im Jahr erscheinen und unter anderem die regelmäßigen Rundschreiben an unsere registrierten User ersetzen. Natürlich soll *die soft.* auch zur Kommunikation und als Forum dienen, für alle, die mit unseren Programmen arbeiten.

Wer Anregungen hat, Tricks und Tips loswerden will oder Kritik vorbringen möchte, wende sich also fortan an: *die soft.* Postfach 10 26 46, D-6900 Heidelberg. Wir sind gespannt, so, wie Sie gespannt sein dürfen.

Die Kunst der ST-Software




Englerstraße 3
D-6900 Heidelberg
Telefon (0 62 21) 30 00 02
Telefax (0 62 21) 30 03 89

Vertrieb in der Schweiz:
Senn Computer AG
Langstrasse 31
CH-8021 Zürich

# *Viren, nochmals Viren*



Virus [*wirus*] das; -, ...ren (kleinster Krankheitserreger)

Ja, so steht es im Duden geschrieben. Dem berüchtigten Computervirus (virus computanti periciosum) kann man seit einiger Zeit auf diversen Rechnersystemen begegnen, d.h. es gibt mittlerweile unzählige davon auf STs, Macs, Amigas, IBMs usw.

Ursprünglich war es eine interne Absprache in der ST-Computer-Redaktion möglichst wenig auf dieses Thema einzugehen, da wir der Meinung sind, je mehr man darüber minuziös schreibt, desto eher werden sie nachprogrammiert und verbessert. Unsere kleine Satire als Editorial der letzten Ausgabe sollte das Thema ausnahmsweise nur kurz anschneiden.

Doch mittlerweile ist bereits bei mehreren Softwarehäuser (zuletzt bei GFA Systemtechnik) dermaßen großer Schaden durch jene besagten Computerviren entstanden, daß wir uns fast schon zwangsweise darüber reden müssen. Als kleiner Tip sei gesagt, daß es im PD-Service (PD 141) bereits ein sogenanntes Antivirusprogramm vorhanden ist, das wir übrigens auch auf jede Einsendung, die unsere Redaktion erreicht, anwenden.

Prinzipiell möchte Ihnen raten, nicht mit einer unbekannten Diskette zu booten, da die meisten Viren im Bootsektor der Diskette angesiedelt sind.

Zuletzt appeliere ich an die ST-Programmierer, keine Viren zu produzieren, da hierdurch großer Schaden entstehen kann. Und wer ein bißchen von der Programmierung des ST versteht, hat es bestimmt nicht nötig, Viren zu produzieren. Das ist für einen halbwegs erfahrenen Programmierer ein Kinderspiel.

Harald Egel

I  

## SOFTWARE

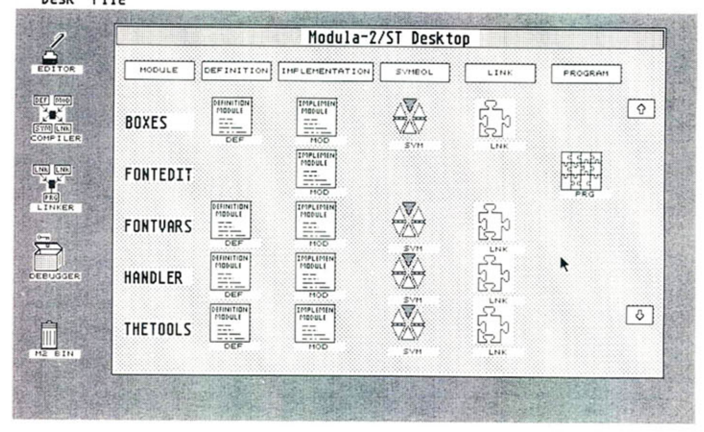


## HARDWARE



## ANWENDUNGEN






### Modula-2 Vergleich

Nachdem wir Ihnen nun im Laufe der Zeit - speziell in den letzten Ausgaben - einige Modula-2-Systeme vorgestellt haben, bringen wir nun in der der Juni-Ausgabe einen großen Vergleichstest dieser neuen Wirth'schen Sprache. Wir hoffen darin, alle Modula-2-Systeme für den ATARI ST zu erfassen (vielleicht gibt es bis dahin schon wieder ein neues) und Ihnen so einen Überblick zu geben.

Seite **28**




### Diskettenformate nach Maß - Diskmaster ST

Sie haben ein hyper-super-duper-megamäßiges, überschlaues Primzahlenprogramm ("42 ist eine Primzahl") im unheimlich angesagten ST-BASIC geschrieben und wollen es jetzt geldbeutelfüllenderweise unter die Leute bringen? Sie suchen dazu noch einen möglichst fiesen Kopierschutz? Dann haben wir was für Sie: DISKMASTER ST, eine bunte Mischung aus Trackmonitor und softer Formatiermaschine, stößt in einen bisher dünn besiedelten Markt vor: Formatierprogramme gibt es zwar mittlerweile mehr als genug, auch Diskmonitore sind keine Mangelware; allerdings beherrschen sie meistens nur ziemlich stinknormale Sektor-Operationen. Wer den Lesekopf seiner Floppy bei der Arbeit bespitzeln will, hat da schon weniger Auswahl.

Seite **38**

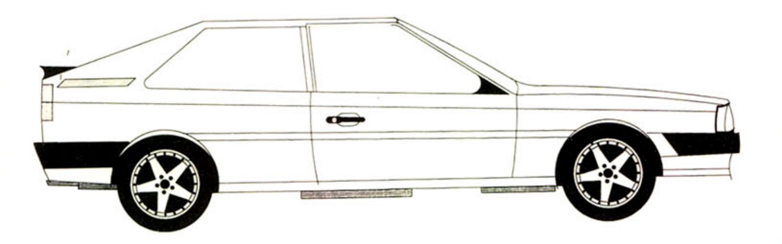
# A L T



## Wenn der ATARI ST mit der Gabi anbandelt

Die Gabriele 9009 von Triumph Adler ist eine der meistverbreitetsten elektronischen Typenradschreibmaschinen. Leider kann sie nur über eine teuere Schnittstellenbox als Computerdrucker verwendet werden. Dieser Artikel gibt einen Einblick in spezielle Druckerfunktionen, stellt das integrierte Schreibmaschineninterface vor und zeigt abschließend eine Softwarelösung auf, die einen direkten Anschluß der Schreibmaschine am ATARI ST doch ermöglicht.

Seite 128



## CAMPUS DRAFT -

Ein preiswertes CAD-Programm mit guten Fähigkeiten Das Einsatzgebiet von CAD-Programmen ist sehr groß. Es reicht von technischen Kontruktionszeichnungen über elektrische Schaltungen und Platinenlayouts bis zum Einsatz im Architekturbereich. Aber auch für die Erstellung von Programmablaufplänen oder Struktogrammen sind sie für Programmentwickler eine große Hilfe. CAMPUS DRAFT ist ein solches CAD-Programm, das im ersten Moment vor allem durch seinen sehr niedrigen Preis auffällt.

Seite 44

## GRUNDLAGEN



## PROGRAMMIERPRAXIS



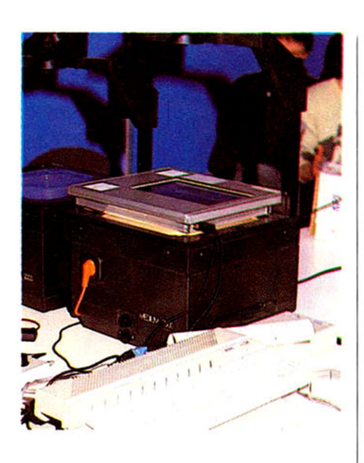
## AKTUELLES



## RUBRIKEN

# NEWS

## LCD-DISPLAY VON CRYSTAL VISION



Leider wurde beim Layout in unserem Messebericht über die CeBIT'88 ein Mixed-Up aus dem OHP LCD-Display von Crystal Vision (richtiges Foto, falscher bzw. kein Text) und dem LCD-Display von Wilhelm Mikroelektronik (falsches Bild, richtiger Text) produziert. Wir bitten dies zu entschuldigen und wollen Sie auch über das Display von Crystal Vision informieren.

Bei dem Gerät handelt es sich um eine innovative Lösung, die den Wunsch nach einem projizierbaren LCD für den ST erfüllt, wobei bei der Konzeption auf ausreichende Sicherheit und Langlebigkeit geachtet wurde.

Da allegemein das zur Abbildung benötigte Polarisationsverfahren bei einer LCD-Anzeige einen großen Teil des Lichtes der Projektionslampe verschluckt, ermöglicht ein spezieller Aufbau im optischen Bereich den Einsatz eines sehr lichtstarken Projektors um diesen Nachteil zu kompensieren, ohne daß dabei eine Beschädigung des LCDs verursacht wird. Auch Projektoren mit Metalldampflampen sind ohne Bedenken verwendbar. Der optische Aufbau des Displays garantiert eine kontrastreiche Abbildung ohne Schlieren, Flecke, und Verschattungen.

Angeboten werden die Formate 640x400 und 640x200, wobei das letztere auch für den IBM PC und alle Portables benutzt werden kann. Der Anschluß zum ATARI ST ist SM124-kompatibel. Die Geräte sind in Supertwisted und Doppel-Supertwisted-Technologie erhältlich. Die betragen DM 5900,- und 2900,-.

Erhältlich sind diese Geräte auch als Direktsichtmonitore mit Hintergrundbeleuchtung oder reflektiv ausgelegt (DM 2400,- bzw. 1400,-).

*Crystal Vision*
*Bernd Haastert*
*Ing.-Labor für Optoelektonik*
*Weberstr. 24 5300 Bonn 1*
*Tel.: 0228/224724*

## Neues ST-BASIC!!!

```
 FILE  FIND  BLOCK  MODE  GO  RUN
 SAVE
 SAVE *.*       tions-Programm für Window- & Menüzeilen-Verwaltung
 LOAD *.*       EMSEL wurde die Library um nicht benötigte GEM-Prozeduren gekürzt
 MERGE *.*
 DIRECTORY      5000
 NEW            MEMORY(32256)+255) AND $FFFF00:Bitblt_Speicher= MEMORY(6+32000)
 QUIT           =0):False= NOT True:Resourcedefs
1030 Appl_Init:Rsrc_Load("GEMDEMO.RSC",X)
1040 IF X<>1 THEN
1050    FORM_ALERT (1,"[3][Ich konnte die Datei|     'GEMDEMO.RSC'|nicht finden.
1060    Appl_Exit: END
1070 ENDIF
1080 Rsrc_Gaddr(0,Menu,Menu_Adr):Menu_Bar(Menu_Adr):Graf_Mouse(0)
1090 Wind_Get(0,4,X0,Y0,W0,H0)
1100 LINE STYLE =1: MODE =3:V_Hide_C
1110 SCREEN 2,Scr2:W=W0-1:H=Y0+H0-1: PRINT "§f";: REM Cursor aus
1120 FOR X=0 TO W
1130    DRAW X,0 TO W-X,H
1140 NEXT X
1150 FOR Y=0 TO H
1160    DRAW W,Y TO 0,H-Y
1170 NEXT Y
1180 SCREEN 0:V_Show_C(0)
```

Ab 1.Juli liefert ATARI Deutschland ein neues BASIC zu den ST-Rechnern mit. So ganz neu ist es allerdings auch wieder nicht, denn es handelt sich um das bekannte OMIKRON.BASIC. Neu ist "nur" der Editor, der ja etwas umstritten war und einige Befehle. Die Verbesserungen können sich allerdings sehen lassen. Allein von der Geschwindigkeit ist der Editor etwa 5 mal schneller geworden.

Er verfügt jetzt über eine Menüleiste, die augenschonend schwarz gehalten ist (s. Bild). Der Editor bildet gleichzeitig eine Art Shell, aus der sich der Compiler und jedes andere Programm aufrufen lassen. Fehler werden jetzt wahlweise blinkend oder nicht blinkend dargestellt. Die kleinste Schriftgröße fällt aus optischen Gründen weg. Blöcke lassen sich per Maus markieren (auch innerhalb einer Zeile). Das Splitten des Bildschirms erfolgt jetzt vertikal (zuvor horizontal), wobei die Anzeigefenster verschieden groß sein können. Die Funktionstastenbelegung und einige zusätzliche Einstellungsparameter sind abspeicherbar.

Drei neue Befehle sind implementert worden. "Find next error" dient u.a. zur leichteren Anpassung von BASIC-Programmen, die mit anderen Dialekten erstellt wurden. "List token" bzw. "Find token" erleichtert die Suche von bestimmten Befehlen, Variablen oder Prozeduren. Mit "Rename token" lassen sich Variablen umbenennen.

Als zusätzliche Libraries gibt es ab Juli eine Inline-Assembler-Library für DM 129,-, die auch mit dem Compiler zusammenarbeitet und eine Numerik-Library für DM 79,-.

*Omikron.Software*
*Erlachstr. 15*
*7534 Birkenfeld*
*Tel.: 07082/5386*

# Neue Version von SEARCH!

Die Firma M. Uhlig pcp hat eine neue Version des bereits bekannten Adreßprogramms SEARCH! angekündigt. Bei dieser Erweiterung des Programms hat man sich weitestgehend nach den eingegangenen Kundenwünschen gerichtet. So wird nun eine sehr schnelle Sortierroutine mitgeliefert, außerdem kann SEARCH! , ohne daß die Menüleiste mit der Maus berührt werden muß, aktiviert werden. Weiterhin nimmt die Kontaktfreudigkeit von SEARCH! zu: Die schon bekannte Möglichkeit, Texte in andere Programme zu übermitteln, schließt auch nicht mehr SIGNUM! aus, da SEARCH! speziell auf diese Textverarbeitung angepaßt wurde. Ein Update für registrierte Benutzer wird zu einem geringen Unkostenbeitrag zu haben sein.
Es ist auch ein kleines neues Programm mit Namen NOLAN entwickelt worden. Hierbei handelt es sich um eine LOW-COST-Diskettenverwaltung, welche zu einem Preis von 24,50 DM erhältlich ist. LOW-COST bezieht sich jedoch nur auf die Verpackung. Dadurch ist es möglich, so der Vertreiber, ein qualitativ hochwertiges Produkt zu niedrigen Preisen zu verkaufen.
Der vielfach bewährte Monitorumschalter ist jetzt auch in einer Kabelversion für MEGA-STs erhältlich. Er wird über ein ca. 50 cm langes Kabel mit dem Rechner verbunden. Zu guter Letzt hat die Firma M.Uhlig pcp ihr Kabel-Sortiment drastisch erweitert, so daß nun alle Anschlußprobleme, die mit Ihrem ST auftreten, gelöst werden. Kostenlose umfangreiche Informationen erhalten Sie bei unter folgender Adresse:

*pcp M.Uhlig*
*Am Hohen Stein 36*
*6200 Wiesbaden*
*Tel.: 06121/65758*

# ST-KONTOR Lager/Fakt

Das Programm ST-Kontor Lagerverwaltung/Fakturierung der ST-Kontor-Programmserie für ATARI-ST-Computer ist ausgeliefert worden. Es ist beim Sybex-Verlag, bei allen Fachhändlern oder direkt beim Hersteller erhältlich.
Das Programm kostet DM 398,- und enthält außer Textbausteinen und Einkaufspreislisten, Selektionen, Umsatzübersichten und betriebswirtschaftlicher Auswertung auch noch eine Bildschirmkasse. Passend zum Programm sind ein Barcodelesegerät, eine Kassenschublade und ein Barcodedruckprogramm erhältlich. An einer eventuellen Erweiterung der Lagerverwaltung um Stücklisten und Aufmaßrechnung wird noch gearbeitet.
Zu allen Programmen der ST-Kontor-Serie werden darüberhinaus vom Hersteller Halb- und Ganztagsschulungen zum Preis von DM 190.- beziehungsweise 390.- angeboten. Dabei handelt es sich um Einzelschulungen im Hause des Herstellers nach Terminvereinbarung.

*Gesellschaft für*
*Finanz- und EDV-Beratung*
*GFE R. Becker KG*
*Königsteinerstraße 76*
*6232 Bad Soden a. Ts.*
*Tel.: 06196/62139*

# Preissenkung für Eickmann Harddisks / Neue Floppies

Zum Juni '88 konnten bei Eickmann Computer die Preise für die angebotenen 40 bzw. 60 MB Harddisks gesenkt werden. So kostet die EX 40-1 mit 40 MB jetzt DM 1.898,-, die EX 60 mit 60 MB DM 2.498,-. Die auf der Basis der SH-205 hergestellten Platten haben nach Herstellerangaben ein gegenüber der SH 205 um 9 dB(A) (!) reduziertes Arbeitsgeräusch, was spürbar angenehmer ist.
Diese geräuschsenkende Maßnahme gibt es auch nachträglich als sogenanntes NR-Kit (Noise Reduction) incl. Einbau für DM 69,-.
Mit der HDPLUS-Version 4.0 von Eickmann-Computer werden u.a. folgende Neuerungen angeboten: Turbo Dos-kompatibel, 8 Partitions pro Laufwerk möglich, Partition kann mit Schreibschutz versehen werden, verbesserte Auto-Boot-Möglichkeit, SubDir-Buffergröße ist frei einstellbar. Außerdem ist das Cache-Speicherprogramm HDCache im Liefer-umfang enthalten. Die Update-Version wird gegen Einsendung der Original-Diskette und eines Verrechnungsschecks über DM 39,- postwendend zugeschickt.
Das von Eickmann Computer angebotene Slot-Kit erlaubt den problemlosen Einbau einer 3 1/2 Zoll-Festplatte in den ATARI PC 1. Mit dem Slot-Kit ist die Nutzung von wahlweise 1 oder 2 kurzen PC-/XT-Karten möglich. Im Lieferumfang enthalten sind Montageplatte, Stromanschluß für Festplatte, Slot-Adapter und Einbauanleitung. Das Eickmann Slot-Kit kostet DM 148,- incl. 20 MB Plattenlaufwerk und Controller DM 948,-.

*Eickmann Computer*
*In der Römerstadt 249*
*6000 Frankfurt/Main 50*
*Tel.: 069/763409*

# JAMES HAT SICH IM PREIS GEIRRT

In der letzten Ausgabe der NEWS haben wir JAMES- den Börsenbutler fälschlicherweise mit einem Preis von 698,- DM angegeben. Leider hat sich JAMES, was bei Butlern selten ist, geirrt. JAMES verkauft sich günstiger und zwar zum Preis von 298,- DM. Weiterhin gibt es einen Zugriff auf die IFA-Datenbank, die eine freie Auswahl von Aktienkursen beinhaltet, bei einem monatlichen Beitrag von 68,40 DM im Monat, allerdings gilt diese Angebot nur für registrierte JAMES-User.

*IFA-Köln*
*Gutenbergstr. 73*
*5 Köln 30*
*0221/520428*

# GFA fängt VIREN

Die Firma GFA Systemtechnik hat bis dato 1400 Exemplare des Buches "GFA BASIC: Version 3.0" ausgeliefert. Unglücklicherweise hat sich auf die Diskette zum Buch ein Virus eingeschlichen - dies wurde leider erst jetzt bemerkt. Obwohl alle Händler ein Schreiben (und ein entsprechendes VIRUS-Killer-Programm) bekommen haben, sind doch einige Bücher schon beim Kunden. GFA Systemtechnik hat nun ein entsprechendes VIRUS-Killer-Programm erstellt, das den Virus nicht nur deaktiviert, sondern auch von der Diskette entfernt.

Wir bitten nun alle Leser, die sich das oben erwähnte Buch gekauft haben, das folgende Listung abzutippen und dieses Programm dann auf die Diskette ihres Buches anzuwenden, um ein weiteres Ausbreiten des VIRUS' zu verhindern.

```
DO
  ALERT 1,"Disk in A: einlegen",1,"Ok",a%
  a$=SPACE$(512)
  DO
    e%=XBIOS(8,L:VARPTR(a$),L:0,0,1,0,0,1)
    EXIT IF e%>=0
    PRINT "Lesefehler ";e%;"   Taste drücken (ESC=Abbruch)"
    IF INP(2)=27
      END
    ENDIF
  LOOP
  IF ASC(a$)=&H60
    PRINT "Diskette infiziert oder bootfähig"
    ALERT 3,"Diskette infiziert|oder bootfähig",1,
                                   "Löschen|Weiter",x%
    IF x%=1
      MID$(a$,1)=STRING$(6,0)
      MID$(a$,59)=STRING$(512,&HE5)
      VOID XBIOS(18,L:VARPTR(a$),L:-1,-1,0)
      REPEAT
        e%=XBIOS(9,L:VARPTR(a$),L:0,0,1,0,0,1)
        EXIT IF e%>=0
        PRINT "Schreibfehler ";e%;
                          "   Taste drücken (ESC=Abbruch)"
      UNTIL INP(2)=27
    ENDIF
  ELSE
    PRINT "Disk sieht normal aus"
  ENDIF
LOOP
```

Nach dieser etwas traurigen Angelegenheit gibt es aber auch positive Neuigkeiten von GFA zu berichten, denn es gilt, zwei neue Produkte in ihrer UTILITY-REIHE anzukündigen: Zum einen stellt sich ein neuer FLOPPY-Speeder vor, der den Zugriff auf die Diskette bis zu einem Faktor 20 beschleunigt und damit eine erhebliche Zeitersparnis bei Programmen schafft, die wiederholt Daten nachladen. Das Programm, dem auch eine resetfeste RAM-Disk beigelegt ist, ist kompatibel zu allen GFA-bekannten Diskettenformaten. Die zweite Utility ist die Multi-Accessory. Wie der Name schon sagt, handelt es sich praktisch um ein stark erweitertes Kontrollfeld, in dem es möglich ist, Parameter in und um den ST einzustellen. Beide Programme sind bei GFA zu einem Preis von 59,- DM zu beziehen.

Zum Thema Preise hat GFA Erfreuliches zu verkünden: Da seit kurzer Zeit GFA-BASIC 3.0 ausgeliefert wurde, kann diese Version auch als Update erworben werden. Wurde die Version 2.0 zu einem Preis von 169,- DM erworben, beträgt die Update-Gebühr 70,-DM, wurden für diese Version 99,- DM bezahlt, hat man für das Update 140,- DM zu bezahlen; ein Update aufgrund der Version 1.0 kostet 90,- DM. Zu beachten ist, daß nur dann ein Update gewährleistet werden kann, wenn das gesamte Paket, das heißt Original-Diskette und (!) Handbuch, mit der Update-Anforderung eingeschickt werden.

*GFA Systemtechnik GmbH*
*Heerdter Sandberg 30*
*D-4000 Düsseldorf 11*
*TEL.: 0211/588011*

# Photo-Publishing auf dem ST

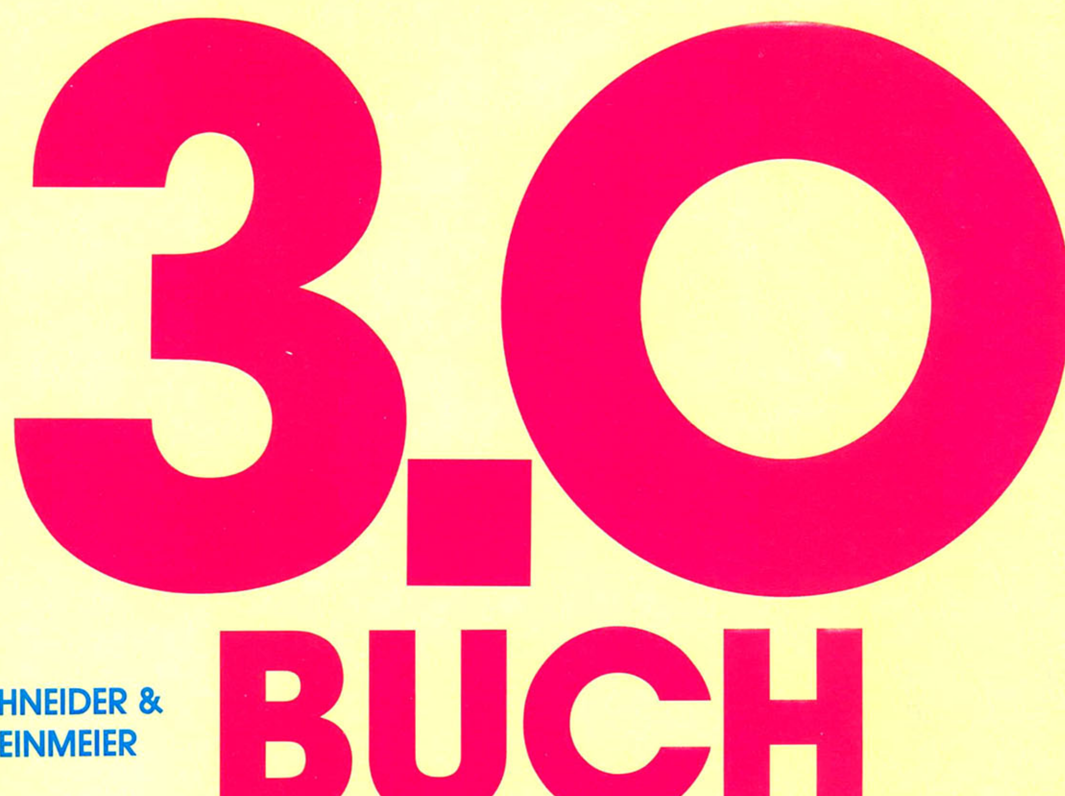
Die Firma Computer Photography hat sich besonders auf die Bildverarbeitung auf Rechnern spezialisiert. Nun stellt Computer Photography ihre ersten vier Programme für den ST vor. Als digitales Photostudio, aber auch als Ergänzung des Desktop Publishings konzipiert, bietet die PHOTO WORKSTATION Anwendungsmöglichkeiten auf dem Gebiet der Bildbe- und verarbeitung. Mit der Version 2.0 erscheinen nun digitalisierte Bilder ohne die üblichen Muster in einer Darstellungsart, die die Ebene der Computerbilder verläßt, da, praktisch unabhängig von der zur Bildverarbeitung benutzten Hardware, die interne Verarbeitung mit 256 echten Grauwerten stattfindet, wobei Bilder, die weniger Graustufen enthalten, automatisch in 256 Graustufen gewandelt werden. Der Bildkontrolle dient eine optimierte Darstellung: Eine unregelmäßige Punktverteilung täuscht die menschliche Wahrnehmung und erzeugt den Eindruck von Grauwerten. Die Ergebnisse können in verschiedenen Formaten abgespeichert werden; auch sind mehrere Druckertreiber vorhanden. Sorgfältig wurde auf maximale Kompatibilität mit der auf dem Markt befindlichen Soft- und Hardware geachtet. Als weiteres Programm stellt sich Slide Maker Plus vor, welches ermöglicht, Schwarzweiß- oder Farb-Diapositive vom Bildschirm herzustellen. Jedes Bildelement kann auf dem Dia eine individuelle Farbe bekommen.

Das Dritte im Bunde heißt BildVision und bietet unzählige Darstellungseffekte von Bildern, die mit beliebigen Quellen erzeugt werden.

Interessant scheint A-Frisur für Friseure und Kunsterzieher zu sein, da es sich um ein Montagesystem handelt, mit dem man Haare ausschneiden, Frisuren sammeln, Gesichtausdrücke variieren und retuschieren kann.

*Computer Photography*
*Stegemühlenweg 48*
*3400 Göttingen*
*Tel.: 0551/7700708*

# DAS GFA-BASIC




# BUCH

**KLAUS SCHNEIDER & OLIVER STEINMEIER**

Erste Tests haben gezeigt, daß das neue GFA-BASIC 3.0 die zur Zeit leistungsfähigste BASIC-Version auf dem ATARI ST ist. Der neue, extrem umfangreiche Befehlssatz erfordert eine grundlegend neue Einführung in die Programmiertechnik. Eine einfache Befehlsübersicht, wie sie mitgeliefert wird, genügt hier nicht.

Als optimale Ergänzung bietet sich dieses Buch an. In zwei Hauptteilen wird zunächst eine systematische Einführung in die Programmierung von BASIC unter Berücksichtigung der besonderen Fähigkeiten des neuen GFA-BASICs gegeben. Hier werden dem Neuling vom ersten Einzeiler bis zu abstrakten Datentypen alle Möglichkeiten der strukturierten Programmierung mit zahlreichen, durch Flußdiagramme transparent gemachten Beispielen nahegebracht. Doch auch BASIC-erfahrene Programmierer lernen hier die neuen Schleifenstrukturen (es gibt noch mehr als FOR, WHILE und REPEAT) kennen.

Der zweite Teil baut auf dem ersten auf und vermittelt weitere Kenntnisse der Programmierung




anhand von Programmen, die wiederum ausführlich beschrieben und erklärt sind. Hier seien eine Fakturierung, eine universelle Random-Access-Datenverwaltung sowie Beispiele zur Betriebssystemprogrammierung, insbesondere von Dialogboxen, genannt.

Durch zahlreiche Anhänge, die neben der obligatorischen ASCII-Tabelle auch einen Index, eine ausführliche Worterklärung sowie weitere nützliche Tabellen enthält, wird das Buch optimal ergänzt.

## AUS DEM INHALT:

**Erklärung der Schleifen- und Programmstrukturen**
- Primzahlenberechnung
- Zahlenraten

**Variablentypen und Arrays**
- Sieb des Eratosthenes
- Adreßeingabe

**Unterprogramme und Prozeduren**
- Rekursionen
- Labyrinthsuche

**Multitasking in GFA-BASIC**

**Abstrakte Datentypen**
- Druckerspooler
- Verkettete Listen
- Binäre Bäume

**Sequentielle Dateiverwaltung**

**Random-Access-Dateien**
- Verkettete Listen auf der Diskette

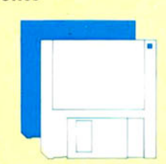
**Grafikprogrammierung**
- Turtlegrafik
- Arbeiten mit mehreren Bildschirmen

**Betriebssystemprogrammierung**
- Aufrufen von TOS-Befehlen
- Verwenden des GEMs
- Menüverwaltung unter GFA-BASIC
- Arbeiten mit Dialogboxen

**Beispielprogramme**
- Fakturierung
- universelle Datenverwaltung
- etc.

**Anhänge**
- Worterklärungen
- Index
- sonstige Anhänge

**Für Einsteiger * Fortgeschrittene * und Profies**



**ÜBER 580 Seiten EINSCHLIESSLICH PROGRAMMDISKETTE**

Dem Buch liegt eine Programm-Diskette bei mit sämtlichen Übungs- und Beispielprogrammen

**BUCH & DISKETTE KOMPLETT DM 59,-**

Preise sind unverbindlich empfohlene Verkaufspreise.

**BESTELLCOUPON**

Bitte senden Sie mir ______ St. DAS GFA-BASIC **3.0 BUCH** einschließlich Programm-Diskette für DM **59,–**
zzgl. DM 5,– Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ______________________

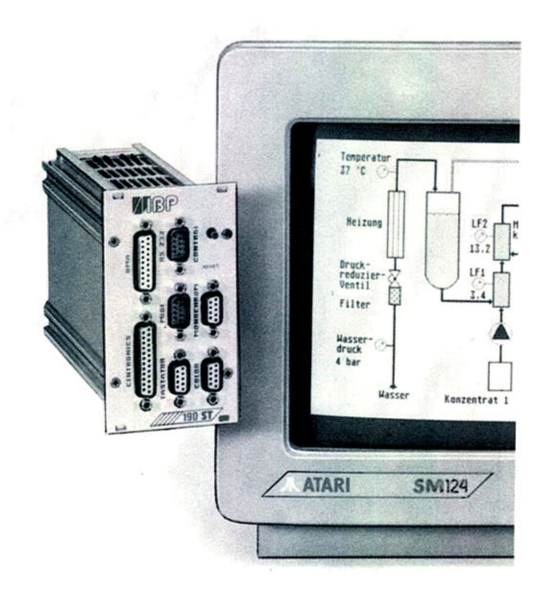
Straße, Hausnr. ______________ PLZ/Ort ______________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

**Heim Verlag**

*Heidelberger Landstr. 194*
*6100 Darmstadt-Eberstadt*
*Telefon 06151-56057*

# 19 Zoll-ATARI im Industrie-Design




Auf der Basis des ATARI ST-Konzeptes und unter Verwendung von ATARI-Bauelementen hat IBP eine Industrieausführung entwickelt. Im Vordergrund der Entwicklung stand die Industrietauglichkeit des Systems. Der Rechner läßt sich durch verschiedene Buserweiterungen vielfältig, z.B. mit AD/DA oder IO/Karte problemorientiert ausbauen. Die Mega - ST -kompatible Hardware ist für 19 Zoll-Montage (3HE/14TE) ausgeführt. Die Zentraleinheit des Systems ist mit einer CPU 68000 ausgerüstet und kann optional um einen mathematischen Co-Prozessor 68881 und einen Blitter (grafischen Prozessor) erweitert werden. Mit bis zu 2 MByte RAM ist das System selbst für anspruchsvolle Aufgaben der Meß-, Regel- und Automatisierungstechnik hervorragend geeignet. Die grafische Auflösung beträgt im Monochrom-Modus 640x400 Punkte,während im Farbmodus wahlweise 320x200 bei 16 Farben oder 640x200 Punkte bei vier Farben zur Verfügung stehen. Das System ist umfangreich mit Schnittstellen ausgestattet, die alle über Sub-D an der Frontplatte zugänglich sind. Neben Centronics-Druckeranschluß sind eine DMA-Schnittstelle mit einer Übertragungsrate von 10MBit/sec., eine Tastatur mit RS232-Pegeln und ST-Protokoll, eine Midi-In/Midi-Out/Midi-Thru, eine RS232-Schnittstelle sowie Schnittstellen zum Color - oder Schwarzweiß-Monitor vorhanden. Die Schnittstellen zu Floppy oder Harddisklaufwerk sind seitlich auf Pfostenverbinder herausgeführt. Daneben sind eine akkugepufferte Echtzeituhr, ein Tongenerator, eine spannungsabhängige Reset-Logik sowie ein Watchdog im System verfügbar. Der Systembus ist wahlweise als EUROBUS-E, MegaST, ECP oder VME-Bus(!!!) auf der hinteren Kassettenseite herausgeführt. Das Betriebssystem MEGA-TOS mit der grafischen Oberfläche GEM wird als Standardsoftware mitgeliefert. Daneben sind andere Betriebssysteme wie z.B. RTOS-UH und OS9 installierbar. Für anwendungsspezifische Software stehen vier Cartridge-Steckplätze zur Verfügung. Das riesige Softwareangebot, insbesondere die vielen verfügbaren Hochsprachen, ermöglichen eine schnelle und kostengünstige Softwareentwicklung. IBP liefert zu ihrer 19-Zoll-Version des ATARI, die übrigens mit einem halben oder zwei Megabyte RAM verfügbar ist, eine Vielzahl an Erweiterungen. Interface-Karten wie ein 16-Kanal-A/D-Wandler, ein 8-Bit-A/D-Wandler mit schaltbaren Effektivwertgleichrichter, eine Ausgangskarte mit 16 Relais- und Optokopplerausgängen, eine Karte mit 24 galvanisch getrennten Eingängen, eine Multi-Timer-Zählerkarte, eine IEEE-Schnittstellenkarte und viele andere sind im Programm vorhanden und bis zum 3.Quartal 88 lieferbar. Weiter werden auch komplette Lösungen und individuelle Software-Entwicklungen für spezielle Probleme bearbeitet.

Auch wenn die Preise sich auf einem anderem Niveau als die des ATARI ST bewegen, so darf man nicht vergessen, daß der ST durch dieses System absolut professionell anwendbar wird. Für Firmen, die den ATARI ST als Meßstation einsetzen wollen, ist dieses Angebot sehr interessant, da sie nun keinen Personal-Computer der Firma ATARI sondern eine Meßstation der Firma IBP kaufen können.

*IBP Gerätebau GmbH*
*Lilienthalstraße 13*
*3000 Hannover 1*
*Tel.: 0511/630963*

# VT52ST-Erweiterung für den ATARI 520 ST

Die Arbeit mit dem ATARI ist teilweise recht mühsam, speicherplatz - und zeitaufwendig. Dagegen steht in TOS-Programmen der VT52-Emulator als einziges Hilfsmittel zur Bildschirmsteuerung zur Verfügung. Dieser enthält allerdings nur einen sehr begrenzten Befehlsumfang. VT52ST V3.4 erweitert den Leistungsumfang des Emulators um zusätzliche Cursor- und Bildschirmoperationen, so daß man auch in TOS- Programmen über eine leistungsfähige Bildschirmsteuerung verfügt. Die Erweiterung des Emulators wurde gewählt, um die zusätzlichen Funktionen möglichst mit jeder Programmiersprache erreichen zu können. Der erweiterte VT52-Emulator zeichnet sich durch schnellere Scroll- und Löschroutinen sowie einen erweiterten Befehlssatz aus. Er läuft mit dem ROM-TOS des 6.2.86 und belegt nur 2650 Bytes.

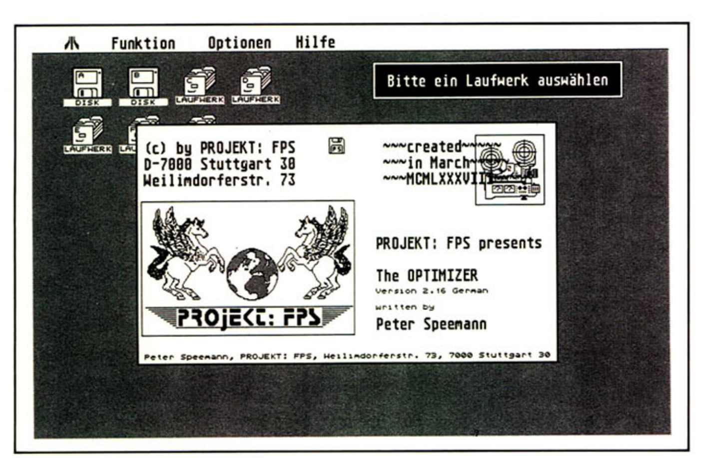
*Siegfried Muschalik*
*Am grünen Platz 25*
*4350 Recklinghausen*
*Tel.: 02361/71327*

# Neuer Flachbett-scanner von WEIDE Elektronik

Unter dem Namen Professional Scanner IX-12 bringt Weide Elektronik für den ST einen neuen Scanner auf den Markt, der auf der Basis eines Canon-Flachbett- oder Trommelscanners arbeitet. Der Scanner verarbeitet in den Auflösungen 300, 200, 150 und 75 dpi und generiert Bilder mit 64, 32 und 16 Graustufen. Die unter GEM geschriebene Software unterstützt die Scan-Modi Degas, IMG und POSTSCRIPT. Dadurch können eine große Anzahl von Programmen wie zum Beispiel Calamus, STAD, DEGAS, die für den ST erhältlich sind , ohne Probleme in Zusammenhang mit dem IX-12 verwendet werden.

*Weide Elektronik*
*Regerstr. 34*
*4010 Hilden*
*Tel. 02103/ 41226*

# The Optimizer - der Harddiskbeschleuniger




Bei "The Optimizer" handelt es sich um ein Programm, das Ihre Speichermedien reorganisiert und dabei, bei regelmäßiger Anwendung, eine Erhöhung der Zugriffsgeschwindigkeit bewirkt. Der Optimizer arbeitet mit allen Speichermedien wie Harddisk, Diskette und sogar mit resetfesten RAM-Disks zusammen. Das Programm zeigt BIOS-Parameter-Blocks, Schachtelungstiefe der Subdirectories, FAT-Sprünge und weitere wichtige Informationen an. Es werden Lost Cluster und Bad Cluster angezeigt, wobei erstere gelöscht werden können, womit man Speicherplatz schaffen kann. Eine Sortierung der Directories ist möglich, um weitere Geschwindigkeitsverbesserungen der Zugriffszeit zu erhalten.

Das Programm, das in drei Geschwindigkeits- und damit Intensivitätsstufen gefahren werden kann, ist, obwohl in Assembler programmiert, sehr anwenderfreundlich gestaltet (siehe Bild). Die Version 2.16 konnte in ihrer Arbeitsgeschwindigkeit gegenüber der Version 2.14 um etwa 98% gesteigert werden.

*Projekt: FPS*
*c/o Peter Speemann*
*Weilimdorferstr. 73*
*7000 Stuttgart 31*

# MOTOROLA kündigt neue digitale Signalprozessoren an !!

Motorola wird ihre bestehende Produktpalette auf dem Sektor der digitalen Signalprozessoren erweitern, die sich erneut durch ihr überaus günstiges Preis-Leistungsverhältnis auszeichnen. Die Bausteine der Baureihe DSP96000 entsprechen in allen Belangen der IEEE-Gleitkomma-Norm 754 aus dem Jahre 1985 und sind voll auf- und abwärtskompatibel zur Familie DSP56000. Echtzeit-DSP-Verarbeitungen, die die IEEE-kompatible Gleitkommaverarbeitung erfordern, profitieren von der 40 MFLOPS (!) betragenden Verarbeitungsleistung. Beispiele für Applikationen dieser Art sind schnelle Steuerungen, digitale Audiosysteme, Bild- und Sprachverarbeitung u.v.m. Der mit 13.33 MIPS arbeitende DSP96001 kann eine komplexe FFT (Fast Fourier Transformation) über 1024 Punkte in weniger als 2ms durchführen. Zeiten, von denen ATARI ST-Benutzer nur träumen können.

Für eine zügige Kommunikation und eine problemlose Speichererweiterung sorgen der jeweils mit 32 Bit herausgeführte Daten- und Adreßbus und die ebenfalls 32 Bit breite Host Schnittstelle. Unter der Bezeichnung OnCETM- Bus besitzen die Signalprozessoren integrierte Hardwarefunktionen und besondere Anschlüsse für die chipinterne Emulation. Dies schafft Voraussetzungen für den Einsatz extrem einfacher und preisgünstiger Hardware für den Entwicklungsprozeß.

Die CPU-Architektur des DSP96002 ist mit derjenigen des DSP96001 identisch, jedoch ist anstelle des HOST-Ports ein zweiter 32Bit-Adreß/Datenbus herausgeführt.

Noch im Sommer werden Entwicklungs-Tools für die neuen Produkte lieferbar sein: die Kombination aus Makro-Crossassembler, Linker und Simulator sowie einem C-Compiler, der in allen Einzelheiten dem Compiler von Kernigham und Ritchie entspricht, und die auf HOST-Systemen wie IBM-PC, Macintosh II, SUN-3 und VAX laufen, lassen kaum Wünsche offen. Entsprechende der MOTOROLA-Strategie auf dem DSP-Sektor werden die Entwicklungs-Tools kostengünstig und einfach zu bedienen sein.

*Motorola GmbH*
*Schatzbogen 7*
*Postfach 82 09 60*
*8000 München 82*
*Tel.: 089/92103-3*

# GDAT bietet neue Tools für APL an

Während microAPL in London immer noch an einer neuen RELEASE für APL/68000-ST arbeitet, das voraussichtlich im Herbst oder Winter erscheinen wird, stellt GDAT ihre neuesten Tools vor.

Da der sehr hohe (aber objektiv gerechtfertigte) Preis des APL-PRINT-WS ein Manko für Käufer unterer Preisschichten darstellt, wurde nun von GDAT eine neue kleine Reihe entwickelt, die sich MINI-Print nennt. Dabei handelt es sich um ein WS mit der Funktion PRINT, die für einen Drucker fest konfiguriert ist und ausschließlich den APL-Zeichensatz beherrscht. Das Programm ist zu einem Preis von 59,- DM erhältlich, wobei auf Anfrage zu erfahren ist, welche Drucker unterstützt werden.

Für Signum II werden, um die Verbreitung von APL-Literatur zu fördern, APL-Zeichensätze vertrieben. Enthalten sind neben der Dokumentation Fonts für Matrixdrucker und Laserprinter sowie zwei DEGAS-Fonts und die Kosten betragen 98,- DM.

APL-EDIT unterliegt weiterhin einem Auslieferungsstop. Es gibt einen nachweislich nicht von GDAT verursachten Fehler im Event-Handling unter APL. ATARI und microAPL streiten sich immer noch um die "Vaterschaft". GDAT hat zwar einen Ausweg gefunden, es müssen aber noch größere Teile der Assembler-Routinen umgeschrieben werden.

Bei GDAT ist auch eine Disk gegen 15,- DM incl. Versandkosten zu beziehen, auf der sich eine Druckroutine für den NEC P6, ein 3-D-Grafikprogramm, Bezier-Kurven, eine Screendump-Routine und vieles mehr befinden.

*GDAT - Gesellschaft für*
*dezentrale Daten/Technik mbH*
*Stapelbrede 39*
*4800 Bielefeld 1*
*Tel.: 0521/ 875 888*

# EcoCheck macht den Firmennamen "wasserdicht"

Bei der Neugründung von Unternehmen sowie der Markteinführung von Produktnamen gehört die Namensfindung zweifellos zu den schwierigsten Kapiteln. In beiden Fällen gilt es zu vermeiden, daß der "neue" Name einem bereits vorhandenen Firmennamen gleicht respektive mit diesem zu verwechseln ist.

Zwar überprüft die zuständige IHK oder HK aufgrund der gesetzlichen Grundlagen Namensvorschläge im Rahmen des jeweiligen Kammerbezirks daraufhin, ob Gleichheit oder Verwechslungsgefahr mit bestehenden Namen vorliegt. Doch wird dies dem Bedarf der Wirtschaft häufig nicht gerecht, da viele Unternehmen überregional tätig sind. Die - gesetzlich vorgeschriebene - überregionale Absicherung konnte bislang nur durch schriftliche Anfragen bei den 440 Registergerichten der Bundesrepublik erfolgen. Neben dieser für den Nutzer unwirtschaftlichen Möglichkeit des Abgleichs - der deshalb in den meisten Fällen unterblieb - stellen auch die vielfach angebotenen Prüfverfahren, die sich beispielsweise an den Daten von Telefon, Adreß- oder Branchenbüchern orientieren, einen unzuverlässigen und darüber- hinaus kostenintensiven Weg dar: Die Datenbestände sind oft veraltet, und es müssen lange Wartezeiten in Kauf genommen werden. Das Ende vom Lied: eine unangenehme und teure Überraschung für den Unternehmensgründer, wenn er einen Firmennamen beim Registerbericht eintragen läßt, die Folgeinvestitionen (z.B. für Werbung, Marketing und Vertrieb) auf diesen Namen ausrichtet, und vielleicht nach einem halben Jahr feststellen muß, daß "sein" Firmenname bereits in einem anderen Kammerbezirk existiert und in dieser Form nicht verwendet werden kann. Muß er dann nach einer Abmahnung den Namen ändern, kann er für einen neuen Namen mit zusätzlichen Investitionen rechnen, die schnell mehrere tausend DM erreichen können.

Problemlos umgehen kann man diese Klippen jetzt mit dem EcoCheck, der erstmals die Möglichkeit bietet, bei einer Namensprüfung auf amtliche und damit zuverlässige Daten zurückzugreifen.

Mit EcoCheck wird ein in der Bundesrepublik bislang konkurrenzloser Dienst zum Namensabgleich angeboten: Aufgrund eines Datenbestandes von mittlerweile über 600.000 Firmennamen überprüft die ECODATA Wirtschaftsinformationen GmbH, Abteilung EcoCheck, beispielsweise Namenswünsche von Unternehmensgründern und Marketingabteilungen großer Konzerne. Der EcoCheck bietet dem Nutzer entscheidende Vorteile:

1.) Der Abgleich der Daten steht auf amtlicher Basis und gewährleistet daher hohe Sicherheit (Grundlage sind die Registereintragungen des "Bundesanzeigers").

2.) Die Kollision bei der Wahl eines Produktnamens mit Firmennamen wird vermieden. Bislang wird vielfach noch übersehen, daß der Inhaber eines Firmennamens unter bestimmten Voraussetzungen von einem Unternehmen verlangen kann, daß dieses ihren dem Firmennamen gleichen oder verwechslungsfähig ähnlichen Produktnamen ändert.

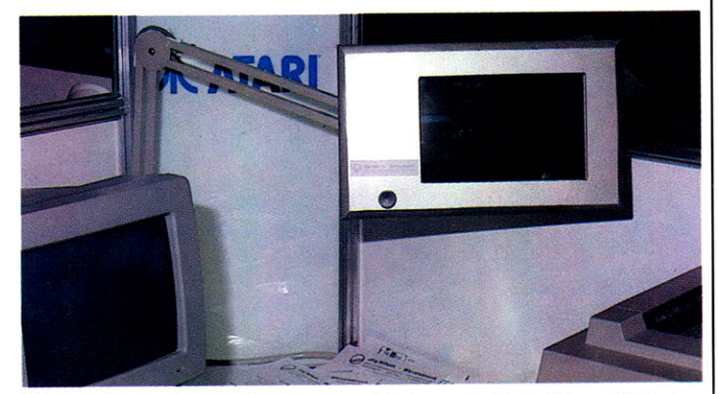
3.) Die Antwort der auf Firmennamensüberprüfungen spezialisierten Fachabteilung von ECODATA erfolgt innerhalb von 24 Arbeitsstunden in der Form von Computerausdrucken oder, auf Wunsch, zusätzlich per Rückruf.

4.) Der Datenbestand, der täglich aktualisiert wird, wächst ständig: In absehbarer Zeit ist damit zu rechnen, daß der Datenbestand vollständig alle registerlich geführten Firmen abdeckt.

Die Kosten für EcoCheck sind, gemessen an den Gesamtinvestitionen für einen Firmennamen außerordentlich gering: Die Überprüfung der ersten zwei Namen kostet ca.350,- DM. Böse Überraschungen und Fehlinvestitioen lassen sich nur durch die rechtzeitige überregionale Namensabsicherung vermeiden!

*ECODATA Wirtschaftsinformationen GmbH*
*Abteilung EcoCheck*
*Postfach 1161*
*8752 Kleinostheim*
*Telefon 0 60 27/60 01*

# LCD-Display von Wilhelm Mikroelektronik




Wie bereits am Anfang der News erwähnt, ist uns ein kleines Mißgeschick bei unserem CeBIT-Bericht passiert. Leider wurde das verkehrte LCD-Display (von Crystal Vision) abgebildet. Nun wollen wir Ihnen auch das richtige Bild zum Gerät der Firma Wilhelm Elektronik liefern.

*Wilhelm Mikroelektronik*
*Süggelstr. 31*
*4670 Lünen*
*Tel.: 02306/14025*

# Es LOHNT sich - Lohn und Gehaltsabrechnung

CIS-Lohn & Gehalt ist ein Programm zur Lohn- bzw. Gehaltsabrechnung, das in kleinen und mittelständigen Unternehmen eingesetzt wird. Das Programm orientiert sich in der Bedienung an der altbekannten Lohn-/Gehaltskarte, wie sie in vielen Betrieben heute noch in Gebrauch ist. Die Aufgabe des Programms besteht jedoch nicht in der Verlagerung der Arbeit vom Papier auf den Computer, sondern vielmehr in der Erleichterung der allmonatlich anfallenden Abrechnung. CIS-Lohn & Gehalt errechnet am Monatsende aus wenigen zusätzlichen Angaben zu jedem Arbeitnehmer automatisch alle abzuführenden und auszugebenden Beträge, bis hin zum Nettolohn, der auf Wunsch direkt an den Arbeitnehmer überwiesen werden kann. Einige Merkmale des Systems sind Auslegung des Programms aufgrund des Einkommensteuergesetzes, wobei ausreichend Spielraum für Sonderbezüge und -abzüge gelassen wurde, Anwählen der Funktionen über Maus und Tastatur und integrierte Lohnsteuertabelle. CIS-Lohn und Gehalt, das zu einem Preis von 1000 DM verkauft wird, wie auch weitere ausführliche Informationen sind bei der Firma Ciechowsky Computer Innovations unter folgender Adresse zu beziehen:

*Ciechowsky Computer Innovations*
*Ober-Saulheimerstr.18*
*6501 Wörrstadt*
*Telefon 06732/7354*

# Neue Version 1.73 des Mailbox-programms ST-Mail

Das Programm, das 398,- DM kostet, bietet aus der Sicht des Sysops und des Benutzer sehr viele Optionen, so daß sie hier nur kurz in Stichpunkten umrissen werden sollen. Weitere Informationen können bei der untenstehenden Adresse direkt angefordert werden.

Aus der Sicht des Sysops geschieht das Arbeiten GEM-unterstützt, wobei Angaben über die Auslastung der Mailbox in den letzten 24 Stunden sowie andere statistische Werte gemacht werden.Das Anlegen von Daten- und Nachrichtenbrettern geschieht in zwei Ebenen, wobei die Zugangsverwaltung bis zu 16 Schlüssel, aber jeder Schlüssel eine beliebige Anzahl von Brettern haben kann. Die Mailbearbeitung ist vielseitig, da Mails in verschiedene Bretter kopiert, gesperrt und freigegeben, sortiert, ediert und ausgedruckt werden können. Jedem User kann eine individuelle System- und Verweilzeit gegeben werden und die Einteilung der User geschieht in zwei groben Kategorien.

Der User wird befehlsorientiert und nicht numerisch geführt. Persönliche Mails, die über einen schaltbaren Fullscreeneditor eingegeben werden können, stehen für jeden Empfänger sofort bereit, wobei sogar Rundschreiben möglich sind. Weiter erwähnenswert sind Funktionen wie Makrodefinitionen, Makroaufruf automatisch oder manuell, Relogin, Xmodem-Übertragung etc.

*Markus Nitschke*
*Software-Entwicklung*
*und -Vertrieb*
*Bahnhofstr. 21*
*6330 Wetzlar*
*Tel.: 06441/48652*

# Hilfsprogramme für Omikron-BASIC

Obwohl Omikron-Basic schon über eine Vielzahl ausgereifter Befehle verfügt, sammeln sich im Laufe der Zeit eine ansehnliche Zahl an Bibliotheken und Hilfsprogrammen. Die Firma EVERYWARE bietet nun ab Mitte Juni eine PROGRAMMER'S DISK ( 35 DM bzw. 40 DM inclusive Nachnahme) an, die etwa folgenden Inhalt haben wird: Programme zum formatierten Ausdruck von Listings, zum Konvertieren von Resource-"H"-Files in BASIC-Dateien, zum komfortablen Auflisten von Binär-Files. Weiterhin werden Libraries zur Behandlung der Fileselect-Box, zur Acccessory-Behandlung, zu Grafik wie zum Beispiel 3-D-Transformation, zum maßstabsgetreuen Koordinatenberechnung, Turtle-Grafik, Bezier-Approximation, Line-A-Einbindung und Einbindung von GEM-Fonts in eigene Programme auf der Diskette zu finden sein.

*EVERYWARE*
*z. Hd. M. Wolf*
*Bodelschwinghweg 15*
*6300 Gießen*
*Tel.: 0641/41450*

# Sample Editor für Prophet 2000/2

Der SAMPLE EDITOR PROPHET 2000/2 (ATARI ST-Version) zeichnet sich, so SYNTEC, durch eine innovative graphische Benutzeroberfläche, die zum Beispiel Sample-Slider und eine modifizierte Fileselect-Box enthält, aus. Die Samples wurden im MIDI-Sample-Dump-Standard integriert. Nach dem Laden, können die Daten beliebig gezoomt angeschaut und bearbeitet werden. Dabei wurden Blockoperationen wie CUT, PASTE, DELETE, APPEND, MIX und diverse RAMP-Funktionen, mit denen zum Beispiel ein ANTI-FADE und Manipulationen der Soundsample-Amplituden linear und exponentiell durchführbar sind, implementiert. Mit diesen Sample-Optimierungen kann der Rauschabstand verbessert und Looppunkte einander angeglichen werden, dadurch sind Effekte wie FADE-IN/FADE-OUT sowie CROSS-FADE möglich. Im Zeichenmodus kann der Benutzer selbst Hand anlegen und seine Soundsamples nach Belieben verändern, optimieren und Effekte erzielen, die durch die eingebauten Funktionen nicht erreichbar wären. Sehr interessant ist, daß man die Samples auch über den Lautsprecher des Monitors wiedergeben kann. Dadurch entfällt beim kreativen Wirken lästige Wartezeit, bis die Daten zum Synthesizer übertragen worden sind.

Das SYNTEC SAMPLE EDITOR P2000/2 ist freigegeben für alle Prophet'en ab der Software-Version 3.0, läuft auf einem ATARI ST mit einem Megabyte-Speicher und SW-Monitor. Das Programm ist ab Ende Juni lieferbar und wird für 298,- DM zu haben sein, wobei die DEMO-Diskette 10 DM kosten wird.

*SYNTEC MUSIC-WARE*
*GREINSTRAẞE 22*
*6100 DARMSTADT*
*TEL.: 06151/375583*

# MÖCHTEN SIE BEI UNS SCHREIBEN?

*Liebe Leser, die ST-COMPUTER gibt es nun schon fast drei Jahre. Die ST-COMPUTER machte sich als erste monatliche ST-Zeitschrift Deutschlands daran, Licht in das Dunkel um das damals noch unbekannte Wesen ATARI ST zu bringen. Viele Geheimnisse dieses Rechners wurden gelüftet und in Form von Programmen oder Artikeln jedem zugänglich gemacht.*

Doch nicht alle Gebiete lassen sich von uns abdecken. Der einzelne Leser ist hierbei gefordert, seine eigenen Erfahrungen und Entdeckungen zu präsentieren. Deshalb der Aufruf an alle Leser:

***Beteiligen Sie sich aktiv an der Gestaltung der ST-COMPUTER.***

Wir möchten Ihnen auf diesen Seiten die einzelnen ST-COMPUTER-Rubriken darlegen und auf einige Besonderheiten verweisen, die Sie wissen sollten, wenn Sie diesem Aufruf folgen.

## PROGRAMMIER PRAXIS

Diese Rubrik ist zur Zeit die beliebteste Sparte unserer Leser. Hier versuchen wir jeden Monat interessante Listings, die ein wenig Licht ins Dunkel um die Programmierkunst des ATARI ST bringen, vorzustellen. Daß hierbei eine aktive Mitarbeit unserer Leserschaft erforderlich ist, können Sie sich sicherlich denken.

Was sollten Sie beachten?
Die PROGRAMMIER PRAXIS enthält Programme oder Routinen zu allen Bereichen der ST-Programmierung. Die Sprache überlassen wir dabei Ihnen. Wichtig ist nur, daß die Listings entweder nützlich oder lehrreich sind.

- Schicken Sie uns den Source-Code und das startbare Programm auf Diskette zu.
- Der Artikel sollte sich ebenfalls auf Diskette befinden. Bei vorhandenem Drucker bitten wir Sie, einen Ausdruck von Text und Listing beizulegen (wenn möglich, in 1st_Word-Format).
- Falls der Artikel Bildschirmfotos benötigt, so sollten Sie die Bilder (als Screendump) auf Diskette abspeichern. Hier sind die Formate beliebig (Doodle, Degas usw.).
- Die Länge des Listings sollte dem Nutzen des Programms entsprechen, d.h., daß man auf Schnörkel, wenn diese das Programm unnötig verlängern, besser verzichtet.
- Wir haben diese Rubrik sinnvoll erweitert. Neben einer Beschreibung der allgemeinen Funktionsweise des Programms (oder der Routine) sollte jetzt ein Struktogramm die Funktionsweise noch besser darstellen. Es soll zusätzlich die Möglichkeit geschaffen werden, daß auch andere, die diese bestimmte Programmiersprache nicht beherschen, Ihr Programm (Routine) in eine andere Sprache umsetzen können. Dieses Struktogramm kann mit jedem gängigen Zeichenprogramm erstellt werden und ebenso auf Diskette eingesandt werden.

Besitzen Sie kein Malprogramm, so schicken Sie uns einfach eine Handskizze, die dann von einem Grafiker hier in der Redaktion "druckreif" gezeichnet wird.
- Bei der Erstellung Ihres Listings beachten Sie bitte,daß es nach Möglichkeit nicht breiter als 55 Zeichen sein sollte.
- Ein Beitrag in der Programmier Praxis muß nicht unbedingt aus einem ablauffertigen Programm bestehen. Auch kurze Routinen, die der Anwender in sein eigenes Programm einbinden kann, sind gefragt. Auf diese Art und Weise lassen sich leicht leistungsstarke Bibliotheken erstellen.

Ihrer Phantasie wird dabei freier Lauf gelassen.
Kennwort: PROGRAMMIER PRAXIS

## PROJEKT

Unter einem Projekt verstehen wir längere Programme (siehe Extended VT52-Emulator oder der neue Linker für das GST-Format), die in einem oder mehreren Heften abgedruckt werden können. Wir benötigen allerdings das ganze Projekt an einem Stück, d.h komplett. Nur dann können wir entscheiden, ob es korrekturbedürfig ist oder noch Erweiterungen nötig sind. Für die Sparte Projekt gilt im allgemeinen das gleiche wie für die PROGRAMMIER PRAXIS. Denken Sie immer daran, daß ein Listing schon mal etwas länger werden kann, aber noch abtippbar und dementsprechend nützlich sein sollte. Den Maßstab für die Länge des Programms können Sie leicht selbst beurteilen. Wieviele Zeilen würden Sie noch abtippen?
Kennwort: Projekt

## HARDWARE

Hardware-Projekte sind in der ST-COMPUTER keine Seltenheit. Wir haben einen Digitizer vorgestellt und erklärt, aus dem Easyprommer haben wir einen weiteren leistungsfähigen Eprommer gebaut, nämlich den Junior Prommer. Unser Umbau des ATARI-Monitors hat eine große Resonanz gehabt und der Scanner war eine preisgünstige Alternative für alle, die mit dem ATARI ST viel im graphischen Bereich anfangen möchten.
Es müssen aber nicht unbedingt sehr große Schaltungen sein. Eine kleine, aber sinnvolle Erweiterung oder ein Zusatzgerät, oder...

Was müssen Sie beachten?
Zu einem Hardware-Projekt gehört außer der Dokumentation (die die Funktionsweise des Gerätes sowie dessen Bedienung erklären sollte) ein Schaltplan (kann auch mit einem Malprogramm erstellt oder als Handskizze beigelegt werden), eine Stückliste und ein Platinenlayout. Bei größeren Projekten muß unbedingt ein lauffähiger Prototyp mitgeliefert werden. Dieser wird nach Begutachtung umgehend zurückgeschickt. Hat das Projekt auch noch einen Softwareteil, so sollte dieser nach den oben genannten Bedingungen (Programmier Praxis sowie Projekt) auf Diskette mitgeliefert werden.
Kennwort: 'Hardware'

## GRUNDLAGEN

Schwerwiegende Themen haben wir bis jetzt in dieser Sparte behandelt. So konnte man in den Floppyspielereien in mehreren Folgen das Innenleben einer "Diskette" verfolgen. Die internen Geheimisse des GEMDOS begleiten Sie zur Zeit immer noch in unserer Serie "Auf der Schwelle zum Licht".
Für gute Grundlagenartikel über den ST und die Welt sind wir jederzeit empfänglich. Themen gibt es genügend. Haben Sie sich in einem bestimmten Bereich erfolgreich hineingearbeitet und wissen darüber zu berichten, so fassen Sie Ihre Erkenntnisse doch einfach in einem Artikel zusammen und schicken ihn uns zu.

Was Sie beachten sollten!
Hier gelten ebenfalls die Regeln, die wir oben genannt haben. Also Text auf Diskette (in 1st_Word-Format), keine Trennung, linksbündig, kein Blocksatz, Bilder auf Diskette.
Kennwort: "GRUNDLAGEN"

## ANWENDUNGEN

Unter diesem Motto möchten wir eine Reihe von praktischen Anwendungen auf dem ATARI ST vorstellen. Den Anfang machte damals der Artikel JuriSTische Anwendungen. Dazu kam die VIP-Serie und "Vom Papier zur Datenbank". Damit dies nicht der letzte Artikel dieser Art bleibt, möchten wir Sie hiermit dazu aufrufen, uns Ihre Erfahrungen und Anwendungen mit dem ATARI ST im alltäglichen Leben mitzuteilen. Bitte schicken Sie uns Ihre Berichte und eventuelle Bilder, die im Beitrag vorkommen auf Diskette.
Kennwort: "ANWENDUNGEN"

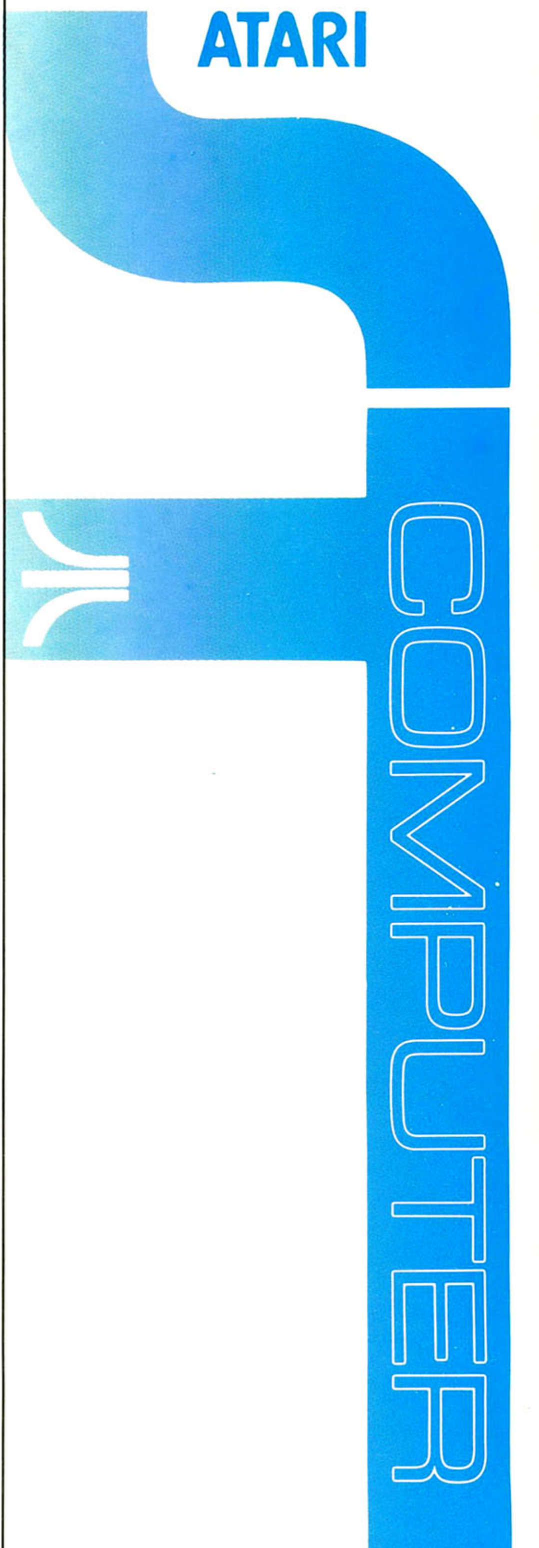
## ST-ECKE

Hier wird ein wenig auf die softwaremäßigen Innereien des Rechners eingegangen. So haben wir zum Beispiel die lange undokumentierten AES-Routinen Appl_tplay und Appl_trecord erklärt. Hier sind also viele wichtige Tips zum Betriebssystem zusammengefaßt. Ansonsten gelten die gleichen Einsendebedingungen wie bei der PROGRAMMIER PRAXIS.

Wir hoffen, daß Sie sich mit dem einen oder anderen Artikeln an der Gestaltung der ST-COMPUTER beteiligen werden. Natürlich wird Ihre Mühe auch honoriert. Ihre gesamten Unterlagen werden zurückgeschickt und die Diskette zurückerstattet. Bei Fragen wenden Sie sich bitte direkt an die Redaktion und fragen nach Herrn Merino, der Ihnen gerne weitere Auskünfte gibt.

Do 14-17 Uhr Tel. 06196/481811

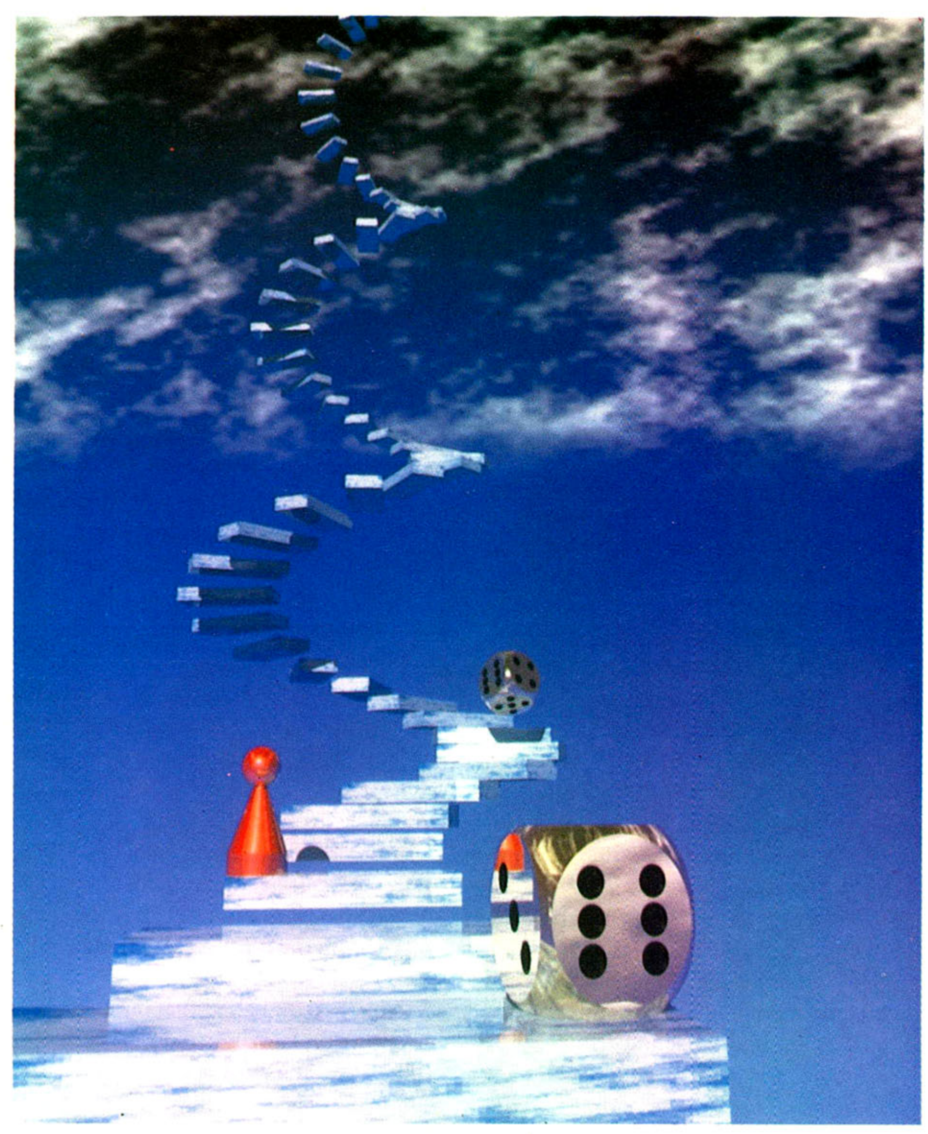
**Kontaktadresse:**
'Merlin'-Computer GmbH
ST-COMPUTER Redaktion
Z.Hd. Herrn Merino
Kennwort: ___________
Industriestr. 26
6236 Eschborn



# BILDERSPIELE

## Ray Tracing - Verfolgungsjagd mit dem Computer

**In den letzten Folgen haben Sie einige Techniken kennengelernt, die benutzt werden, um schattierte Bilder darzustellen. Immerhin, weiche Farbverläufe und Schlaglichteffekte sind möglich. Man kann diese Verfahren mit erheblichem Aufwand auch noch weiter verbessern, so daß Schatten und Transparenz möglich werden. In dieser Folge soll aber nun ein schon vom Konzept her völlig anderes Prinzip der Berechnung realistischer Bilder vorgestellt werden, das zwar einen erheblich höheren Grundaufwand erfordert als die 'traditionellen' Verfahren, dafür aber all die schönen Spezialeffekte gleich in einem Aufwasch miterledigt. Dazu sind Ray-Tracing-Programme auch noch relativ leicht zu implementieren. Vorhang auf.**





Das Ray Tracing-Verfahren wurde ursprünglich als Hidden-Surface-Verfahren entwickelt. In dieser Form tut ein Ray-Tracing-Programm also nichts anderes als die in der letzten Folge vorgestellten Verfahren, wie der z-Buffer oder der Maleralgorithmus. Allerdings ist ein Ray-Tracing-Programm als reiner Hidden-Surface-Prozessor extrem ineffizient.

Schon sehr schnell stellte sich aber heraus, daß das Verfahren viel mehr Möglichkeiten bietet. Heute sind Ray Tracing-Systeme das Standardwerkzeug in der High-End-Computergrafik. Kein Verfahren erlaubt es, so gute Bilder zu berechnen. Einziger Nachteil ist der hohe Rechenaufwand. Aber auch da versucht man, mit immer neuen Verbesserungen Abhilfe zu schaffen.

Die Grundidee ist einfach. Der Betrachter einer beliebigen Szene sieht Licht, das von irgendwelchen Lichtquellen (Lampen, Sonnen, Feuer, Kerzen usw.) ausgesandt und von den Gegenständen, aus denen die Szene besteht, in Richtung des Betrachters reflektiert wird. Über die einfache Modellierung des Verhaltens von Licht haben Sie ja bereits in der letzten Folge dieser Serie einiges erfahren. Auch bei einem Ray Tracer gilt: Je besser das Lichtmodell, desto besser die Ergebnisse. Ray Tracer erlauben übrigens den Einsatz erheblich besserer Lichtmodelle als die 'traditionellen' Verfahren.

Um festzustellen, was der Betrachter sieht, verfolgt man einfach jeden Lichtstrahl von seinem Entstehungsort an und beobachtet, wo er hinstrahlt, wohin er reflektiert wird usw...

Oder nein, vielleicht ist das keine allzugute Idee, denn von den Lichtstrahlen, die eine Lichtquelle aussendet, erreicht nur ein verschwindend kleiner Teil den Betrachter. Außerdem sendet so eine Lichtquelle unendlich viele Lichtstrahlen aus, was zu einer etwas unpraktisch langen

Rechenzeit führen könnte. So geht es also nicht. Aber umgekehrt... Wenn man einfach vom Auge des Betrachters aus alle Lichtstrahlen in die Szene verfolgt, hat man schon mal entschieden weniger zu tun. Aber immer noch unendlich viel. Man muß sich also noch etwas einfallen lassen, das die Anzahl der zu verfolgenden Strahlen begrenzt. Dazu fällt uns sofort der Bildschirm ein. Schließlich müssen wir das Bild sowieso auf einem endlichen Raster aus z.B. 640*400 Punkten darstellen. Also verfolgen wir nur diejenigen Strahlen, die vom Betrachterstandpunkt. (Wir nehmen selbstverständlich einen punktförmigen Betrachter an.) aus durch ein Pixel des Bildschirms auf die Szene fallen. Um die Rechnerei zu vereinfachen, nehmen wir einfach mal an, der Bildschirm läge auf der xy-Ebene. In Bild 1 sehen Sie, wie das aussieht. Jeder Strahl muß sich irgendwie mathematisch beschreiben lassen, ohne Steckbrief können wir ihn schlecht verfolgen. Zwei Punkte des Strahls sind uns bekannt, die Koordinaten des Pixels, dessen Strahl wir gerade verfolgen und die Koordinaten des Blickpunktes. Daraus läßt sich glücklicherweise eine Geradengleichung konstruieren, und zwar derjenigen Geraden, die durch das Pixel und den Blickpunkt geht.
Es gilt:

$$x = x1 + (x2 - x1)*t$$
$$y = y1 + (y2 - y1)*t$$
$$z = z1 + (z2 - z1)*t$$

wobei x,y,z ein beliebiger Punkt auf dem Strahl ist, x1,y1,z1 der Blickpunkt und x2,y2,z2 das entsprechende Bildschirmpixel. Da wir die xy-Ebene für den Bildschirm gewählt haben, ist z2 gleich 0. Wenn t=0 ist, erhält man als Ergebnis der Gleichung den Blickpunkt, wenn t=1 ist, ergibt sich das Pixel. Daraus kann man entnehmen, daß t>1 wird, wenn wir den Strahl weiter in die Szene verfolgen.
Warum tun wir das überhaupt? Nun, wir wollen wissen, welches Objekt vom Blickpunkt, wenn er sozusagen entlang des Strahls durch das Pixel schaut, sichtbar ist. Dieses Pixel müssen wir dann nämlich entsprechend dem gefundenen Objekt verändern. Finden wir ein blaues Objekt, muß dieses Pixel blau werden, ist es rot, wird das Pixel eben auch rot. Um festzustellen, auf welches Objekt der Strahl trifft, berechnen wir einfach Schnittpunkte zwischen dem Strahl und allen Objekten der Szene. Wenn der Strahl kein Objekt trifft, verläßt er die Szene und das Pixel erhält die Farbe des Hintergrundes (z.B. des blauen Himmels

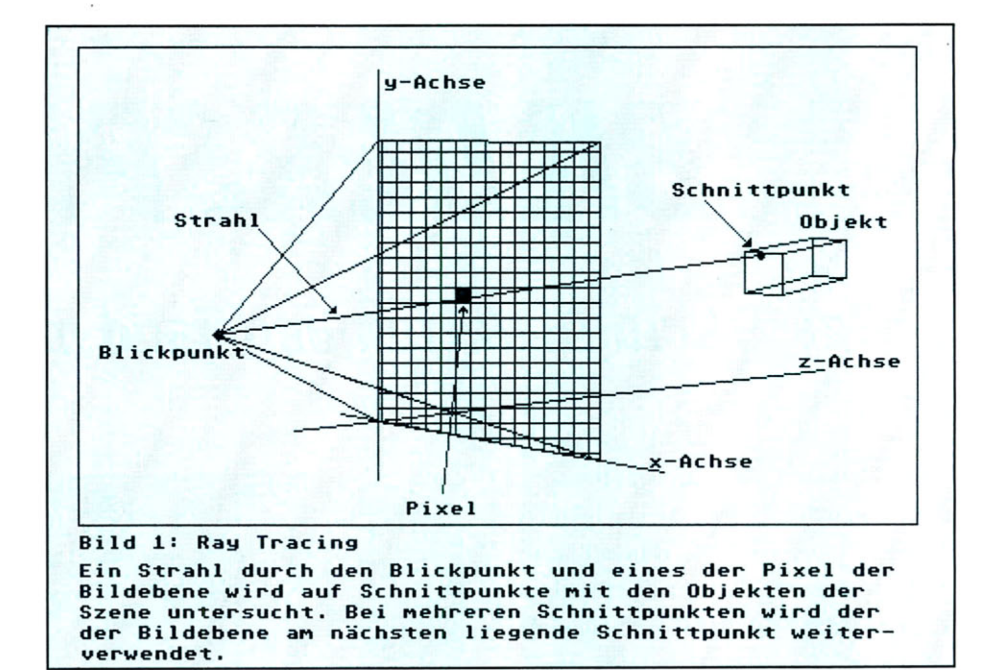


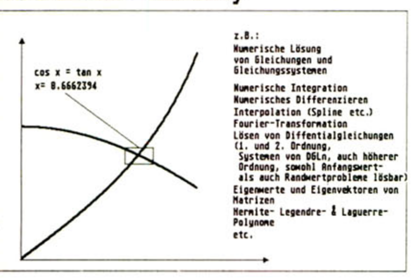
**Bild 1: Ray Tracing**
Ein Strahl durch den Blickpunkt und eines der Pixel der Bildebene wird auf Schnittpunkte mit den Objekten der Szene untersucht. Bei mehreren Schnittpunkten wird der der Bildebene am nächsten liegende Schnittpunkt weiterverwendet.

über dem Kornfeld). Trifft er nur ein Objekt, ist alles klar. Gibt es aber Schnittpunkte mit mehreren Objekten, müssen wir wissen, welcher Schnittpunkt dem Beobachter am nächsten liegt, denn dieses Objekt verdeckt dann alle anderen. Das ist einfach. Am Pixel ist der Wert für t gleich 1, am Blickpunkt 0. Wenn sich jetzt für die Schnittpunkte Werte für t zwischen 2 und 7 ergeben, liegt der Punkt mit dem kleinsten Wert für t dem Bildschirm (und damit auch dem Beobachter) am nächsten.
Wenn alle Schnittpunkte aller Strahlen (ein Strahl durch jedes Pixel) mit jeweils jedem Objekt berechnet worden sind, ist der Algorithmus am Ende (Ihre Geduld wahrscheinlich auch). Für jedes Pixel wurde jetzt festgestellt, welches Objekt von dort aus sichtbar ist. Dies ist die einfachste Aufgabe des Ray Tracing-Verfahrens, nämlich die Auffindung aller sichtbaren Objektteile.
Vielleicht ist Ihnen aufgefallen, daß dies im Prinzip eine Digitalisierung der dreidimensionalen Welt ist. Eigentlich treffen unendlich viele Strahlen den Beobachter, Sie wählen jedoch nur eine endliche Anzahl davon aus. Nichts anderes geschieht beim Digitalisieren von Musik, z.B. für die CD-Produktion; das kontinuierliche Musiksignal wird in einzelne Abtastwerte zerlegt. Wie für Musik ist auch hier das Abtasttheorem von Shannon gültig, nach dem sich immer dann Verzerrungen ergeben, wenn die Abtastrate niedriger als das Doppelte der höchsten im abzutastenden Signal vorkommenden Frequenz ist. Was man bei Musik hören kann, sieht man auch auf dem Bildschirm: leider sind diese Verzerrungen, die sich bei Rasterbildschirmen als häßliche Treppchen in schrägen Konturen zeigen, nur mit aufwendigen, Anti-Aliasing genannten Verfahren zu mildern. Anti-Aliasing ist eines der kompliziertesten Gebiete der digitalen Signalverarbeitung, deshalb habe ich diesen Punkt in dieser Serie auch fein säuberlich ausgespart. Es ist gar nicht so einfach, ein Anti-Aliasing-Verfahren zu implementieren, das akzeptable Ergebnisse bei vernünftigem Aufwand ergibt.
Noch etwas anderes. Wie berechnet man Schnittpunkte eines Strahles mit einem Objekt ?
Ganz einfach, man benötigt dafür nur eine mathematische Beschreibung des Objektes. Eine Kugel ist mathematisch sehr einfach zu beschreiben, das Männchen bei der Juggler-Demo, die wohl jeder Amiga-Besitzer kennt oder sein eigen nennt (ha, das reimt sich !), ist z.B. aus Kugeln zusammengesetzt. Es ist auch möglich, frei geformte Oberflächen mathematisch zu beschreiben, aber dann steigt die ohnehin schon lange Rechenzeit eines Ray Tracers stark an, weil Schnittpunkte mit solch komplexen Objekten sehr schwer zu bestimmen sind.
640*400 Pixel sind insgesamt 256.000 Pixel. Wenn eine Szene aus zehn Objekten, was wahrlich nicht viel ist, besteht, müssen 256.000 Strahlen mit jeweils 10 Objekten geschnitten werden. Das macht 2.560.000 Schnittpunktberechnungen. Und das sind nur die Schnittpunkte, die für die Hidden Surface-Berechnung gebraucht werden. Alle Effekte, die realistische Bilder erzeugen, haben wir ja noch gar nicht erwähnt.
In Bild 2 sehen Sie die Berechnung des

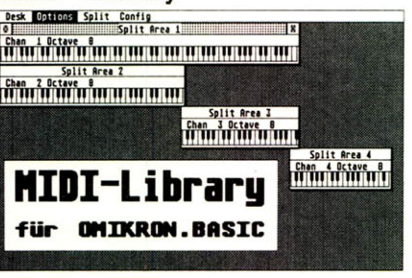
**STATISTIK – Library**

Umsatz 87/88
Auswertung der Telefonaktion
Mai Juni Juli Aug Sep Okt Nov Dez Jan

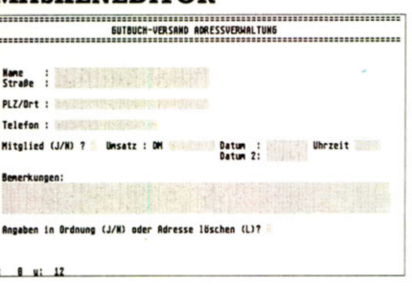
**NUMERIK – Library**




**MIDI – Library**




**MASKENEDITOR**




**D.R.A.W.!**






# OMIKRON.BASIC: das Entwicklungssystem

„Der OMIKRON.BASIC-Interpreter stellt zusammen mit dem Compiler das beste BASIC-Entwicklungssystem für den Atari ST dar." (ST Computer 9/87, S. 64)

## Der Interpreter

Ein umfangreicher Befehlssatz, ein leistungsstarker Editor und höchste Ausführungsgeschwindigkeit kennzeichnen den OMIKRON.Interpreter.

- 19 Stellen Genauigkeit bei allen Funktionen
- Matrizen-Algebra
- Masken-INPUT, SORT-Befehl (mit deutschen Umlauten)
- Strukturiertes Programmieren: rekursive Prozeduren, mehrzeilige Funktionen
- 99 % MBASIC-kompatibel

OMIKRON.BASIC auf Diskette **DM 179,–**
OMIKRON.BASIC auf Modul **DM 229,–**

## Der Compiler

Noch mehr Tempo erreichen Ihre in OMIKRON.BASIC geschriebenen Programme durch unseren OMIKRON.Compiler.
Der Compiler erzeugt ein Assemblerprogramm (echte Maschinensprache, keinen P-Code!), das direkt vom Desktop aus angeklickt werden kann. Auch Accessories können Sie mit dem OMIKRON.Compiler erstellen. Durch CUTLIB werden nur diejenigen Ausführungsroutinen eingebunden, die tatsächlich gebraucht werden; die Folge sind außerordentlich kompakte Programme. Die Ablaufgeschwindigkeit der compilierten Programme ist außergewöhnlich. Der Tester in der mc meinte gar: **„Nur Assembler ist schneller"** (mc 2/87, S. 122)

OMIKRON.Compiler **DM 179,–**
OMIKRON.Compiler für 68881
(vollkompatibel zum normalen Compiler) **DM 229,–**

## Die Libraries

Wer professionell Programme entwickelt, möchte nicht jedesmal das Rad neu erfinden. Mit neuen Libraries (Befehls-Bibliotheken) und Hilfsprogrammen macht OMIKRON.BASIC Ihnen das Leben jetzt noch etwas leichter. Libraries erweitern den Befehlssatz: Sie laden einfach die Library, und die neuen Befehle stehen Ihnen sofort zur Verfügung.
Außer den links abgebildeten Programmen gibt's noch eine Robot-Library (ab Mai) und einen **I**ntegrierten **D**ebugger **E**ditor **A**ssembler **L**inker (kurz **IDEAL**), mit dem Sie Maschinenprogramme erstellen und austesten können (ST Magazin 19, Seite 74: „Debugger der Superklasse")

Alle links abgebildeten Programme **je DM 79,–**
IDEAL **DM 99,–**

Gerne schicken wir Ihnen nähere Informationen zu – Anruf genügt!




**Programme für Programmierer**


**OMIKRON.SOFTWARE**
Erlachstraße 15 · D-7534 Birkenfeld · ☎ (0 70 82) 53 86

Frankreich: OMIKRON · France, 11 Rue Dérodé, F-51100 Reims
Luxemburg: ELECOMP, 11 Avenue de la gare, L-4131 Esch-Alzette
Niederlande: Terminal Software Publicaties, Postbus 111, NL-5110 Baarle-Nassau
Österreich: Ueberreuter Media, Laudongasse 29, A-1082 Wien
Schweiz: Thali AG, Industriestrasse 6, CH-6285 Hitzkirch

Schnittpunktes eines Strahles mit einer Kugel. Mit Zylindern ist es auch nicht viel schwieriger. Die Grundidee bleibt immer die gleiche; man setzt die Gleichung des Strahles in die des Objektes ein und schaut nach, ob etwas Sinnvolles herauskommt. Wenn nicht, gibt es keinen Schnittpunkt.

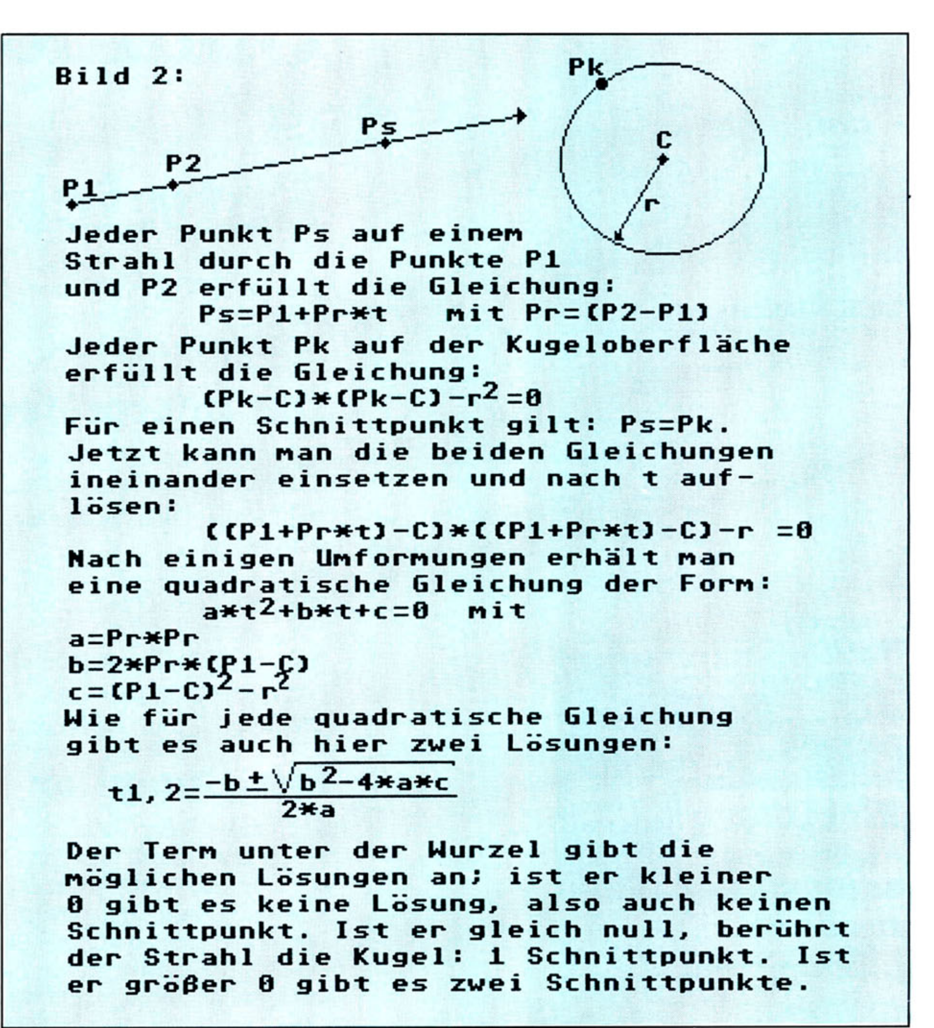
## Special Effects...

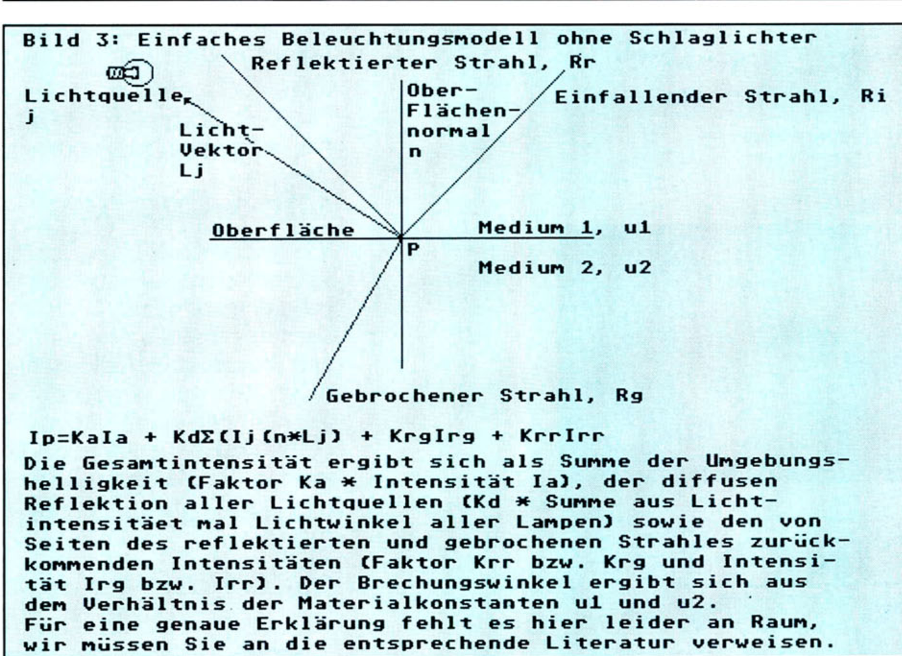
Bisher haben wir einfach aufgehört, wenn wir ein erstes Objekt gefunden hatten. So geht das aber nicht. Das Licht kommt ja nicht von diesem Objekt, sondern von irgendeiner Lichtquelle und ist nur von diesem Objekt reflektiert worden. Die geometrische Optik liefert alle Gesetze, die man zur Berechnung der Lichtbrechung auf einer Oberfläche braucht. Was wir tun müssen, ist festzustellen, unter welchem Winkel ein Lichtstrahl die Oberfläche erreicht. Damit kann man dann auch feststellen, aus welcher Richtung er die Oberfläche wieder verläßt. Wir tun dies natürlich andersherum.

Wir wissen ja, wie der Lichtstrahl gekommen sein müßte und verfolgen also den Weg des Lichtstrahls über alle Ecken und Kanten, bis er die Szene verläßt oder auf eine Lichtquelle trifft. Damit wissen wir, welche Intensität er gehabt hat, als er losgesendet wurde. Wenn er von der Lampe kommt, hat er die Intensität der Lampe, kommt er aus dem Hintergrund, die der Umgebungshelligkeit. Bei jeder Reflektion wird diese Intensität gemindert, wobei die Stärke der Minderung von den Oberflächeneigenschaften des reflektierenden Objektes abhängt. Ein Spiegel mindert wenig, eine matte Oberfläche stark.

Wenn man transparente Oberflächen zuläßt, taucht ein weiteres Problem auf. An einer transparenten Oberfläche wird ein Strahl nicht nur gespiegelt, sondern auch gebrochen (Bild 3). Auch hierbei gelten wieder die Gesetze der geometrischen Optik. Der erste Schnittpunkt ist im allgemeinen also nicht das Ende der Strahlverfolgung. Im Gegenteil, hier spaltet der ursprüngliche Strahl sich in einen Baum von reflektierten und gebrochenen Strahlen auf, die alle verfolgt werden müssen, bis sie die Szene verlassen.

Erst dann, wenn die Ausgangspunkte aller Strahlen, die aus dem Ursprungsstrahl entstanden sind, berechnet wurden, kann damit begonnen werden, die tatsächlichen Helligkeitsverhältnisse am ersten Schnittpunkt zu berechnen. Die




Helligkeit an dieser Stelle ist nämlich die Summe aller auf den Punkt auftreffenden Teilhelligkeiten. An jeder Brechungs- oder Reflektionsstelle müssen, vom Ursprungsort ausgehend, die Veränderungen, die die Objekte je nach Materialeigenschaften dem Lichtstrahl angetan haben, berechnet werden. Die Summe all dieser Teilintensitäten ergibt die endgültige Helligkeit des Pixels.

Die Berechnung der Schnittpunkt-Intensitäten wird mit Beleuchtungsmodellen wie denen, die Sie bereits in der letzten Folge kennengelernt haben, durchgeführt. Wie bereits oben erwähnt, gibt es aber erheblich raffiniertere Simulationen des Lichtverhaltens. Sogar die Reflektion von diffusem Licht zwischen verschiedenen Objekten und die diffuse Streuung an Nebel, Staub oder Rauch kann mit einem

komplexen Modell und einer dazugehörigen Abwandlung des Ray Tracing-Verfahrens (sogenannte Radiosity-Methode, entwickelt an der Cornell-Universität) berechnet werden. Ergebnis sind ganz besonders subtile Licht- und Schatteneffekte. In diesem Fall liegt der besondere Gewinn im Detail.

Bei einem Ray Tracer genügt es nicht, das Lichtmodell für jede Grundfarbe einzeln zu berechnen. Im allgemeinen müssen sogar für jede Grundfarbe eigene Brechungen und eigene Strahlbäume berechnet werden - nicht jedes Material reflektiert und bricht das Licht in der gleichen Richtung, ein Glasprisma beispielsweise bricht jeden Lichtanteil unterschiedlich, Ergebnis ist ein Farbspektrum. Aus Gründen der Rechenzeit verzichtet man allerdings meistens auf derartige Feinheiten.

Jetzt fehlen nur noch die Schatten, die über unseren Häuptern schweben... Sie stellen aber überhaupt kein Problem dar. An jedem Schnittpunkt konstruiert man außer den gebrochenen und reflektierten Strahlen sogenannte 'Shadow Feeler', Schattenfühler. Das sind Strahlen, die vom Schnittpunkt aus in Richtung auf jede vorhandene Lichtquelle zeigen. Diese Strahlen müssen jetzt wiederum mit jedem Objekt geschnitten werden. Liegt kein Objekt zwischen Schnittpunkt und Lichtquelle, ist der Schnittpunkt von dieser Lichtquelle voll beleuchtet. Liegt ein nichttransparentes Objekt dazwischen, liegt der Schnittpunkt, auf diese Lichtquelle bezogen, im Schatten. Ist das Objekt dazwischen transparent, entsteht ein Halbschatten, die Wirkung der Lichtquelle auf den Schnittpunkt muß gemindert werden (Bild 4).

Sie sehen, die Anzahl der Schnittpunktberechnungen hat sich inzwischen vervielfacht. Für jede Reflektion und jede Brechung müssen wieder Schnittpunkte der entstehenden Strahlen, für jeden Schnittpunkt vor Anwendung des Lichtmodells Schattenfühler für jede Lichtquelle berechnet und auch diese mit jedem Objekt geschnitten werden. Bei Szenen mit vielen Objekten kann das leicht zu 100 (oder auch erheblich mehr) Schnittpunktberechnungen pro Pixel führen. Und die sind bei komplexen Objekten auch erheblich komplizierter als bei Kugeln. Kein Wunder also, daß man auch nach Einsatz aller Beschleunigungstechniken immer noch davon ausgeht, das 800-1000 MFlop für die Real-Time-Berechnung von Ray Tracing-Bildern benötigt werden. Selbst die schnellsten heute verfügbaren Computer benötigen mindestens 10 Minuten für ein Bild, an dem ein ST oder Amiga Jahre zu rechnen hätte.



Copyright 88 by ArtCom GmbH, Atelier für Computergrafik BITZ, Bremen

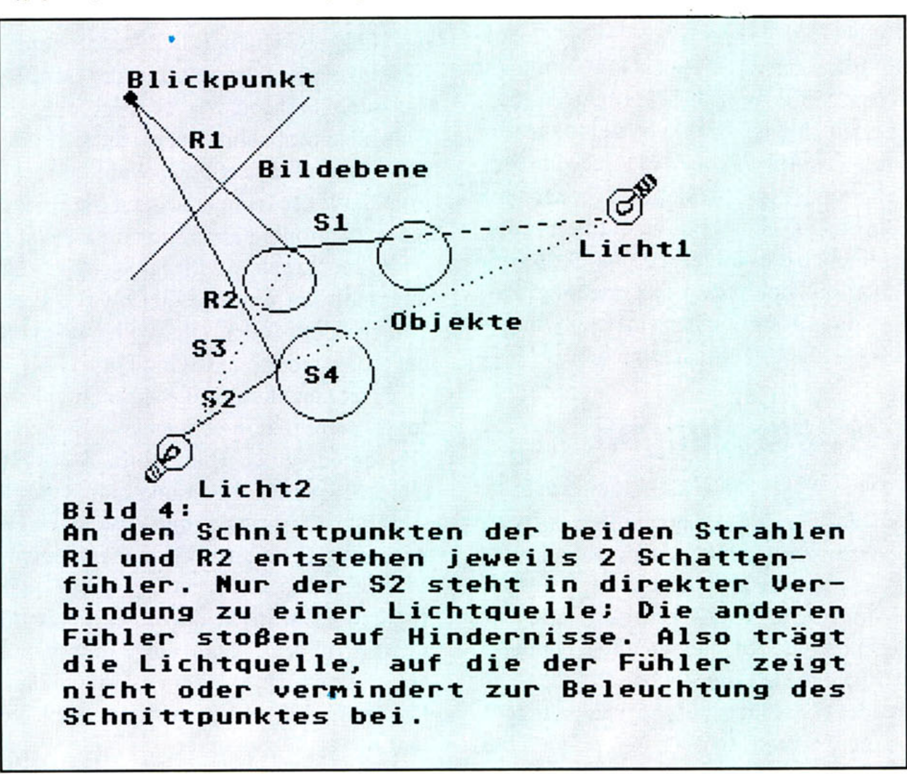



Bild 4:
An den Schnittpunkten der beiden Strahlen R1 und R2 entstehen jeweils 2 Schattenfühler. Nur der S2 steht in direkter Verbindung zu einer Lichtquelle; Die anderen Fühler stoßen auf Hindernisse. Also trägt die Lichtquelle, auf die der Fühler zeigt nicht oder vermindert zur Beleuchtung des Schnittpunktes bei.

Dennoch, der Aufwand lohnt sich. Keine andere Technik kann so realitätsnahe Computerbilder erzeugen wie eben Ray Tracing. Und bei wirklich komplexen Szenen ist es auch um die Effizienz der anderen Techniken, die von Objekten und nicht vom Bildschirm ausgehen, nicht mehr so gut bestellt.

Schließlich fällt, wenn man die Fachliteratur verfolgt, auf, daß viele Möglichkeiten oder Feinheiten, die im Labor implementiert sind, in kommerziellen Systemen vernachlässigt sind. Bald werden also noch viel leistungsfähigere Grafiksysteme zur Verfügung stehen.

Es sei noch erwähnt, daß es natürlich auch möglich ist, einen Ray Tracer für Objektbeschreibungen aus Polygonen, wie sie in den Verfahren der letzten Folgen aufgezeigt wurden, zu verwenden. Die Berechnung von Schnittpunkten mit Polygonen ist sogar besonders einfach, dabei ist aber zu beachten, daß selbst einfache Objekte im allgemeinen aus

sehr vielen Polygonen zusammengesetzt sind, so daß sehr viele Berechnungen nötig werden. Wenn Sie also solche Objekte verwenden wollen, werden Optimierungsverfahren, wie sie im nächsten Abschnitt beschrieben werden, noch viel wichtiger, als für 'einfache' Objektstrukturen wie zum Beispiel mathematische Beschreibungen.

Es gibt im Falle der Verwendung von Polygonen noch etwas zu beachten: Wie Sie bereits aus der letzten Folge wissen, haben Polygone nur eine einzige Oberflächennormale. Und mit den üblichen Lichtmodellen erhalten diese Polygone auch mit einem Ray Tracer immer nur eine einzige Schattierungsintensität. Wie langweilig. Um also eine vernünftige Schattierung zu erreichen, muß, wie beim Phong-Schattierungsverfahren, für jedes Pixel eine Normale interpoliert werden. Die Normale wird auch für die Berechnung der Lichtbrechung und Reflektion benötigt. Polygone erfordern also besondere Rücksicht.

Natürlich haben sie auch ihre Vorteile. Wenn Sie Polygonbeschreibungen verwenden, können Sie ein System konstruieren, das Animationen in Real Time mit Drahtmodellen, schnelle Skizzen mit Gouraud- oder Phongshading und schließlich die endgültige Produktion der Bilder mit einem Ray Tracer ermöglicht. Und das alles mit einer einzigen Objektbeschreibung. Praktisch, nicht?

## Verbesserungen

Das Wort 'Beschleunigungstechniken' im vorigen Absatz hat Sie hoffentlich neugierig gemacht. Es wurden, gerade in neuester Zeit, einige Algorithmen entwickelt, die Ray Tracing erheblich effizienter machen, und vor allem den fatalen Effekt, daß die Rechenzeit mit steigender Objektanzahl exponential ansteigt (Warum wohl?) auf mindestens lineare Verhältnisse mildern.

Eine Analyse von Ray-Tracing-Programmen hat ergeben, daß das Verfahren über 95% seiner aktiven Rechenzeit mit Schnittpunktberechnungen verbringt. Nahezu alle Optimierungen beschäftigen sich daher mit zwei Fragen: 1. Wie kann man die Schnittpunktberechnungen beschleunigen? und 2. Wie kann man die Zahl der Schnittpunktberechnungen reduzieren?

Eine Beschleunigung der Schnittpunktberechnungen ist nur in den seltensten Fällen durch verbesserte mathematische Verfahren möglich. Zum Beispiel wurden die Verfahren für die Schnittpunktberechnung mit parametrischen, freiformbaren Oberflächen in den letzten Jahren stark verbessert. Dennoch liegt das Hauptgewicht bei allen Verbesserungen auf einer Verringerung der Anzahl der Berechnungen.

Bild 5:
Ein Baum aus Strahlen wird von oben nach unten aufgebaut. An jedem Knoten werden die neuen Verzweigungen berechnet. Wenn alle Strahlen die Szene verlassen, wird der Baum in umgekehrter Reihenfolge durchgearbeitet; an jedem Knoten wird das Beleuchtungsmodell berechnet, wobei in das Ergebnis die Werte der bereits berechneten Strahlen eingehen (Siehe Bild 4: Die Summanden KrgIrg und KrrIrr). Das Endergebnis an Knoten 1 ist der Wert, der dem Pixel, durch das der Strahl führt, zugeordnet wird.

Eine erste Maßnahme ist es, die Umrisse der darzustellenden Objekte auf konventionelle Weise rechnerisch auf die Bildebene zu projizieren. Dann müssen nur noch die Strahlen verfolgt werden, die innerhalb der Umrisse liegen. Bei sehr komplizierten oder sehr vollen Szenen nützt das leider praktisch nichts.

Sehr gebräuchlich ist die Zusammenfassung mehrerer beieinander liegender Objekte in einer Hierarchie. Beispiel: Objekte sind ein Tisch und eine daraufstehende Obstschale mit Äpfeln. Jetzt kann man diese ganze Gruppe mit einem Volumen umschließen, einer Kugel oder Box etwa. Ein Strahl, der dieses Volumen nicht schneidet, kann auch unmöglich eines der Objekte im Inneren schneiden. Die Obstschale wird ebenfalls samt Inhalt in ein Testvolumen gepackt. Wenn nun ein Strahl die Box der gesamten Szene schneidet, kann man wiederum die Box um die Obstschale testen. Auch hier gilt: Wenn der Strahl die Box nicht trifft, kann er auch keines der Objekte im Innern treffen. Dieses Verfahren nutzt Zusammenhänge innerhalb des Bildes aus und kann dann einige Erleichterung schaffen. Problematisch ist unter Umständen die Auswahl eines vernünftigen Umhüllungsvolumens: es muß einfach zu berechnen sein (am besten eine Kugel), aber auch möglichst gut passen, sonst nützt es nicht viel.

Besonders die Polygonrepräsentierung, bei der ja jedes Objekt aus vielen Einzelpolygonen besteht, für die jeweils ein Schnittpunkt berechnet werden muß, gewinnt viel durch dieses 'Bounding Box'-Verfahren.

Es gibt ein weiteres Verfahren, das erstmals in einem Artikel vom letzten Jahr ('87) beschrieben wurde. Es nutzt den Umstand aus, daß entsprechend dem Verhalten natürlicher Objekte jedes Pixel den umliegenden Pixeln verhältnismäßig ähnlich ist. Beispiel: Wenn Sie eine Fläche bearbeiten, werden alle Pixel, außer den am Rand liegenden, nur leichte Änderungen gegenüber den umliegenden Pixeln aufweisen. Die Gemeinsamkeiten zwischen den Bildteilen eines Objektes lassen uns dieses Objekt erst als solches erkennen. Diese Eigenschaft kann man zu einer Vorsortierung aller möglichen Strahlen nutzen, womit sich ein annähernd konstantes Verhalten des Algorithmus erreichen läßt. Einer der bestechenden Vorteile des einfachen Ray Tracers, seine simple Implementierung, geht damit aber verloren.

Eine ganze Reihe von Verbesserungen basieren auf einem anderen Prinzip. Hier wird der an sich kontinuierliche, 'analoge' Raum in einzelne Raumzellen zerlegt, ähnlich wie ein Rasterbildschirm die Fläche des Bildschirms durch ein Raster aus Pixeln darstellt. Hier gibt es zwei Ansätze:

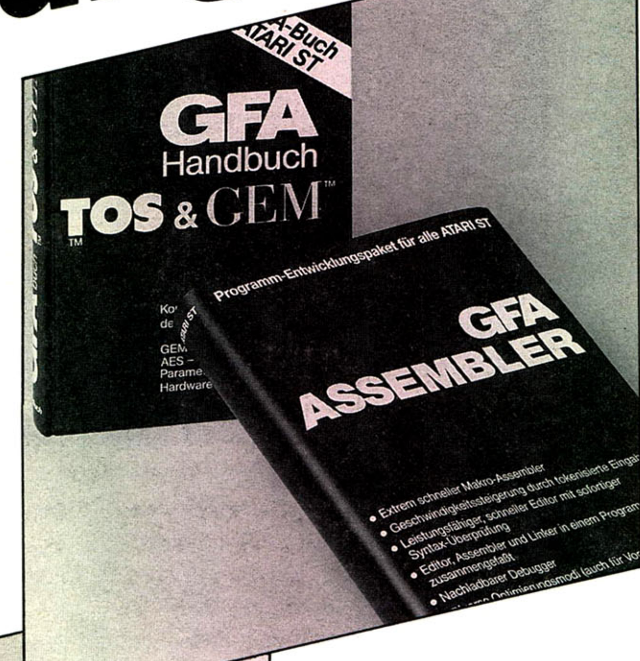
Der meines Erachtens interessanteste ist ganz neu und basiert auf einer Zerlegung des Raums in gleichmäßig große Raumzellen, exakt den Pixeln des Bildschirms entsprechend. Zum Verfolgen eines Strahles wird jetzt auch keine mathematische Berechnung verwendet, sondern ein Verfahren, das eine dreidimensionale

# Software-Leistung

## Für Profi-Programmierer

**GFA-ASSEMBLER**
Extrem schnelles Programm-Entwicklungspaket für alle ATARI ST. Editor, Assembler und Linker sind in einem Programm zusammengefaßt, der Debugger ist nachladbar.
DM 149.-

**GFA-Handbuch TOS & GEM**
Das Nachschlagewerk für jeden ST-Programmierer. Das Buch enthält eine komplette Übersicht sämtlicher Routinen und dokumentierter Variablen des Betriebssystems.
ISBN 3-89317-002-2
368 Seiten + 32 Seiten Indexheft
DM 49.-




## GFA-UTILITY-Serie

**Farbkonverter**
Stellt die geringe und mittlere Auflösung auf dem hochauflösenden Monitor dar. Konvertiert die meisten Farb-Anwender-Programme. Spiele werden in der Regel nicht konvertiert.
DM 59.-

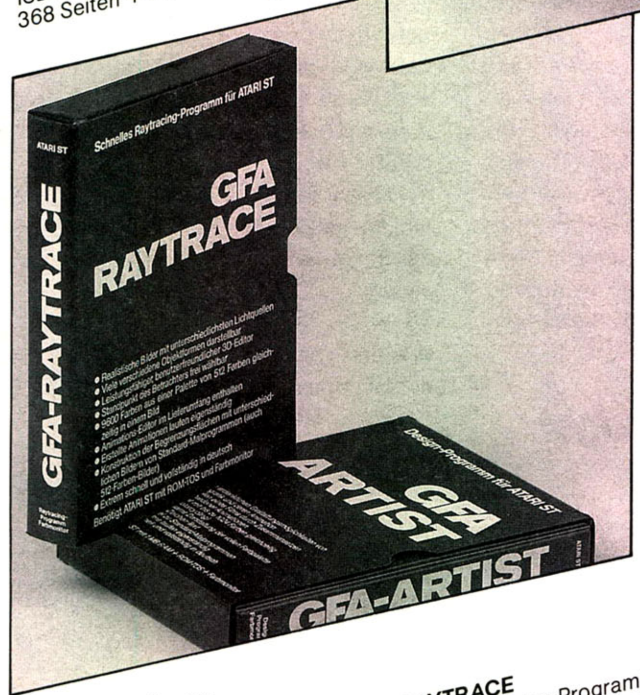
**Monochrom-Konverter**
Die hohe Auflösung ist mit diesem Utility auf dem Farbmonitor darstellbar. Die meisten Monochrom-Programme werden konvertiert.
DM 59.-

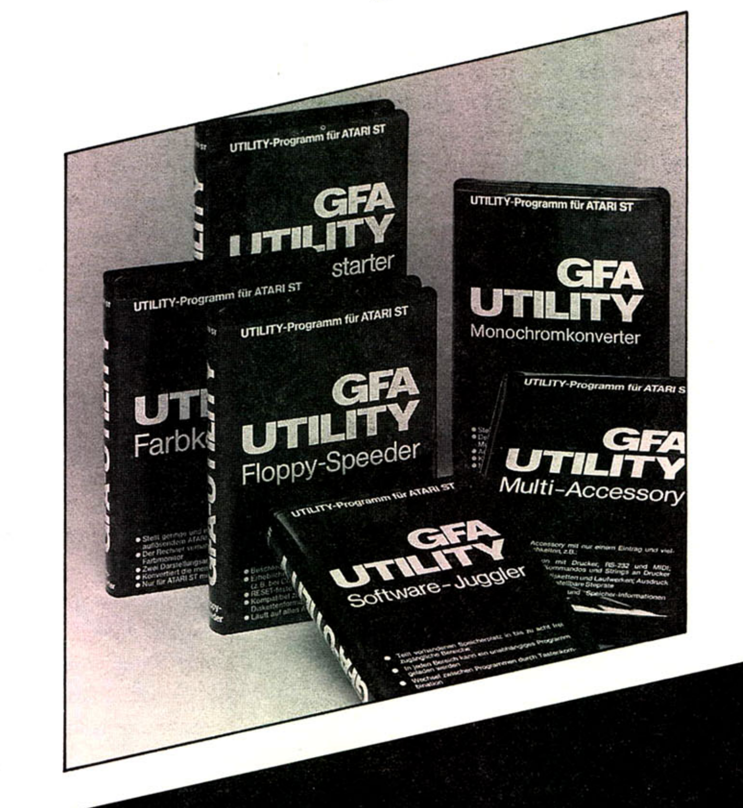
**Floppy-Speeder**
Diskettenoperationen werden bis zu Faktor 20 beschleunigt. Die RESET-feste GFA-RAM-Disk ist im Lieferumfang enthalten.
DM 59.-

**GEM-Autostarter**
Bis zu 10 Programme können hintereinander aus dem Auto-Ordner gestartet werden. Die Übergabe von Parametern ist möglich.
DM 59.-

**Software-Juggler**
Teilt vorhandenen Speicherplatz in bis zu acht frei zugängliche Bereiche. In jeden Bereich kann ein unabhängiges Programm geladen werden.
DM 59.-

**Multi-Accessory**
Universelles Accessory mit nur einem Eintrag, jedoch vielseitigen Möglichkeiten für die tägliche Arbeit mit dem ATARI ST.
DM 59.-




## Animation

**GFA-ARTIST**
Design-Programm für alle ATARI-ST mit 1 MB RAM, ROM-TOS und Farbmonitor. Alle erdenklichen Gestaltungsmöglichkeiten von Farbgrafik und deren Animation stehen Ihnen zur Verfügung.
DM 149.-

**GFA-RAYTRACE**
Schnelles Raytracing-Programm. Darstellung und Animation von realistischen Bildern mit unterschiedlichsten Lichtquellen. Benötigte Hardware: ATARI ST mit ROM-TOS.
DM 149.-

GFA Systemtechnik GmbH
Heerdter Sandberg 30
D-4000 Düsseldorf 11
Telefon 02 11/58 80 11



Erweiterung eines Algorithmus' zum Linienziehen im Zweidimensionalen darstellt. Dieses Verfahren hat den Vorteil, daß ausschließlich Integer-Arithmetik - nicht eine einzige Division ist nötig - verwendet werden kann. Der Strahl wird einfach als Linie im 3D-Raum in die Raumzellen hineingezogen. Wenn er auf eine nicht leere Zelle stößt, ist dies automatisch der Schnittpunkt mit dem nächsten Objekt, der Strahl pflanzt sich ja vom Bildschirm aus beginnend in seiner Richtung fort. Damit spart man sich langwierige Schnittpunktberechnungen mit Objekten, die am Ende den Strahl gar nicht berühren. Die Geschwindigkeitsgewinne des Verfahrens sind enorm: Bilder, die auf traditionelle Weise auf einer VAX vorsichtig geschätzt 40 Tage Rechenzeit benötigt hätten, waren nach dem neuen Algorithmus in zweieinhalb Stunden fertig. Man muß allerdings zwei Schritte beachten: In Experimenten hat sich gezeigt, daß man die Zahl der Raumzellen sorgfältig wählen muß. Zwischen 40*40*40 und 50*50*50 Zellen haben sich als optimal erwiesen. Für jede Raumzelle muß, bevor der eigentliche Tracer seine Arbeit aufnimmt, eine Liste angelegt werden, in der steht, welche Objekte in der Zelle enthalten sind. Wenn der Tracer dann später eine Zelle erreicht, muß er nur kontrollieren, ob die Zelle leer ist. Wenn ja, darf er weiter machen, andernfalls muß er Schnittpunkte nur mit den in der Liste eingetragenen Objekten berechnen.Effizient ist dieser Algorithmus natürlich nur bei komplexen Szenen. Der Grundaufwand ist so hoch, daß bei Szenen mit sehr wenigen Objekten ein traditioneller Ray Tracer vergleichbare Effizienz aufweist (Warum wohl???).

Der andere Ansatz zerlegt den Raum in eine Hierarchie von verschieden großen Zellen. Leere Raumgebiete werden zu großen, leeren Zellen, andere Gebiete werden solange unterteilt, bis die Zellen klein genug sind, um eine einfache Beschreibung des Objektteiles zu ermöglichen. Die Strahlverfolgung kann auch in dieser Variante mit einem 3D-Linienalgorithmus erfolgen, es geht aber auch anders. Die Resultate dieses Verfahrens sind nicht ganz so günstig, wie die des vorigen und zudem von der Kohärenz der Szene abhängig. Auch für bestimmte, hochkomplexe Objekttypen wie frei formbare Oberflächen oder fraktale Landschaften gibt es Verfahren, die die bei diesen Objektformen extrem aufwendigen Schnittpunktberechnungen stark beschleunigen.




Copyright 88 by ArtCom GmbH, Atelier für Computergrafik BITZ, Bremen

Eine letzte Gruppe von Verbesserungen befaßt sich mit der Bildqualität. Auch mit Ray Tracing in seiner Grundform ist es nicht möglich, das gesamte physikalische Verhalten von Licht zu simulieren. Neuere Ergänzungen setzen sich zum Ziel, auch die Lichtbeugung an einem Spalt oder die Diffusion in nebelartigen Athmosphären berechnen zu können. Aber dies geht weit über diesen Artikel hinaus. Bei all diesen Veränderungen und Verbesserungen sieht man den ursprünglich so einfach zu implementierenden Algorithmus mit Tränen in den Augen zu einem hochkomplexen Monstrum werden...

## Implementierungshinweise

Mit ein paar Hinweisen zur Implementierung soll dieser Teil seinen Abschluß finden.

Es bietet sich förmlich an, den Algorithmus so aufzubauen, das Strahlen auf einen Stack abgelegt werden. Vor dem Aufruf der eigentlichen Tracer-Routine berechnet ein init-Teil den Ausgangsray für ein bestimmtes Pixel, schiebt ihn auf einen Stack und ruft dann den Tracer auf. Dieser holt einen Strahl vom Stack und berechnet Schnittpunkte. Wenn er den nächsten gefunden hat, berechnet er die entstehenden Strahlen und speichert auch diese auf dem Stack. Dann beginnt er von vorne und wiederholt diesen Vorgang solange, bis alle entstehenden Strahlen ins Leere gelangen.

Jetzt geht es umgekehrt: Der Reihe nach werden die Strahlen vom Stack geholt und für ihre Schnittpunkte, die natürlich in der Datenstruktur 'Ray' gespeichert sein müssen, wird das Beleuchtungsmodell angewendet. Das Beleuchtungsmodell sollte die Schattenfühler berechnen und damit sein Ergebnis beeinflussen, das zur Helligkeit des jeweiligen Pixels addiert wird.

Wenn der Stack leer ist, ist man wieder am Ausgangspunkt angelangt und kann das Pixel auf die berechnete Helligkeit setzen.

Sie sehen also, daß es überhaupt nicht schwierig ist, einen primitiven Ray Tracer zu implementieren. Allerdings rechtfertigt die Bildqualität eines ST die dafür notwendige Rechenzeit meines Erachtens nicht so ganz. Ein Gouraud-Shading-Algorithmus mit Transparenz, Schatten und Texturemapping ergibt ähnliche Ergebnisse in weit kürzerer Zeit. Aber wer Lust hat, sollte sich doch einmal an eines der Beschleunigungsverfahren machen...

## Großes Finale

Meine Damen und Herren, dies ist nun das Ende dieser Grafikserie. Falls Sie irgendwelche Fragen oder besonderen Wünsche zu diesem Thema haben, schreiben Sie uns. Wir werden uns dann bemühen, in einer Fortsetzung diesen Vorschlägen entgegenzukommen. Ich hoffe, Sie haben einen kleinen Einblick in das weite Feld der Computergrafik ge-

wonnen. Zum Schluß möchte ich Ihnen noch ein neues Buch vorstellen, das gerade im Springer Verlag erschienen ist. Es ist wohl die vollständigste Zusammenstellung von Algorithmen für die komplizierteren Probleme der Computergrafik, die bisher erschienen ist. Es beinhaltet von grafischen Datenstrukturen bis zur Simulation natürlicher Phänomene wie Wasser, Feuer oder Wolken, von Lichtmodellen über Methoden zur Berechnung von Schatten und Transparenzeffekten über Beschleunigungstechniken für Ray Tracer einfach alles. Wenn Sie Interesse an Computergrafik haben, müssen Sie dieses Buch einfach lesen. Es ist gut illustriert und erklärt, ein extrem vollständiges Verzeichnis von Originalliteratur rundet das Werk ab. Insgesamt fast 400 Seiten voll mit geballtem Wissen. Ach so, den Titel und die Verfasser habe ich fast vergessen:
**Image Synthesis - Theory and Practice** von den in der Szene sehr bekannten kanadischen Professoren **Nadia Magnenat-Thalman** und **Daniel Thalmann**. Herausgeber ist der ebenfalls nicht unbekannte **Tosiyasu L. Kunii**. Erschienen ist das Buch in der Springer-Serie 'Computer Science Workbench'. Leider ist es nicht ganz billig, 148.- DM kostet es, aber wert ist es dieses Geld auf alle Fälle.
So, mit diesem Tip für Fans verabschiede ich mich endgültig von Ihnen. Tschüß,

*Christian Schormann*

*ENDE*

# MODULA-2 HAT AUFGEHOLT

## - Alle Compiler im Vergleich

**Für den Atari ST gibt es inzwischen viele ernstzunehmende Entwicklungssysteme. Ein Schwerpunkt liegt - "ST-historisch" begründet - bei den C-Compilern. Im Laufe des letzten Jahres kamen erstaunlich viele Compiler für eine andere Sprache auf den Markt: Modula-2.**

In diesem Vergleich sollen die bisher getesteten Pakete gegenübergestellt werden. Detailaspekte, auf die hier nicht eingegangen wird, finden Sie in den Tests der vergangenen Monate beschrieben.

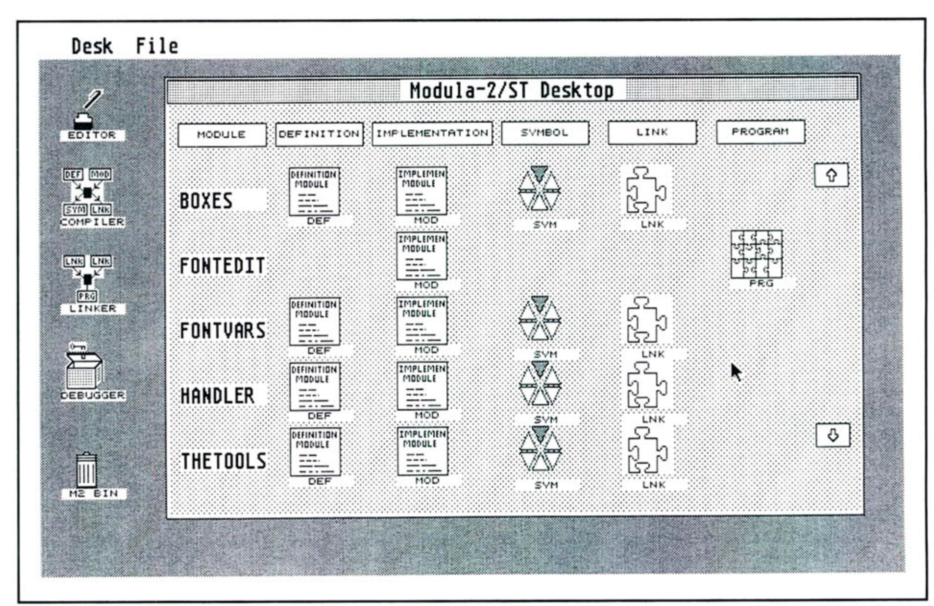



*Bild 1: Die TDI-Oberfläche*

## Modula-2, etwas Besonderes?

Modula-2 ist der Nachfolger von Pascal (das sich auf dem ST erstaunlicherweise nicht durchsetzen konnte) und ebenfalls von Niklaus Wirth entworfen. Ein zentrales Konzept der Sprache ist die Modularisierung. Programme werden nicht an einem Stück notiert oder in Include-Dateien aufgeteilt, sondern mit Hilfe von Modulen auch logisch zerlegt. Dabei stellt ein Modul der Außenwelt Objekte wie Prozeduren oder Typen zur Verfügung. Ein Modul, das ein anderes importiert kann diese Objekte benutzen, ohne deren Implementierung zu kennen. Eine Änderung der Implementierung einer importierten Routine verändert das importierende Programm nicht. Solange die Schnittstelle gleich bleibt, muß lediglich neu gelinkt werden. Der Compiler überprüft die Einhaltung der Schnittstelle, vor allem Typübereinstimmungen, sehr penibel.
Gegenüber Pascal ist Modula-2 auch für systemnahe Programme geeignet. Auch wenn es von vielen bestritten wird: In Modula-2 kann man Tricks wie in C anwenden. Allerdings müssen sie expliziert notiert werden und verschwinden nicht in einem unlesbaren Dschungel von Sonderzeichen. Der Programmiererspruch aus USA: "Modula-2's the best language for hacking" trifft diesen eher unbeachteten Aspekt von Modula-2 haargenau.
Ein weiteres Merkmal von Modula-2 ist unter den auf Mikros weiter verbreiteten Sprachen einzigartig: die Nebenläufigkeit. Als Teil der Sprache sind Standardfunktionen definiert, die das Umschalten von einer Koroutine in eine andere enthalten. Damit lassen sich Programme schreiben, in denen verschiedene Prozeduren quasi-parallel an einem Problem arbeiten. Bei Koppelung der Umschaltung an einen Timer ist Multitasking per Definition in Modula-2 vorhanden.
Weiterhin sorgen viele Features der Sprache für lesbare und übersichtliche Programme. Die Eidgenössische Technische Hochschule in Zürich (ETH) bietet einen Compiler zur Portierung an. Eine Modula-Implementierung für den ST ist also eigentlich ein Kinderspiel.
Trotzdem sind die für den Atari ST angebotenen Compiler sehr unterschiedlich, und auch nicht alle beruhen auf dem ETH-Stammvater. In diesem Vergleich werden alle uns zugänglichen Systeme vorgestellt und die jeweiligen Besonderheiten herausgestrichen.

## TDI Modula

Der TDI-Compiler kam als erster auf den ST-Markt und war lange Zeit das einzige Paket. Er liegt inzwischen in der Version 3.0 vor und wurde auch schon auf den Amiga portiert. Daher ist er sicherlich der fehlerfreieste Compiler und hat bewährte und korrekte Bibliotheken. Die Version 3.0 ist ein Fünf-Pass-Compiler. Es soll auch eine neuere Revision als Ein-Pass-Übersetzer nach dem neuesten

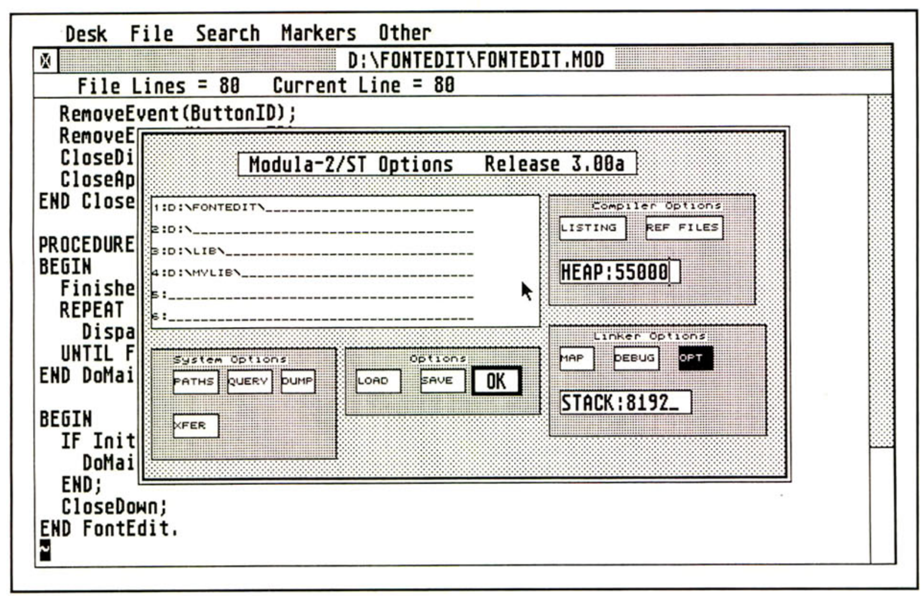



*Bild 2: Sämtliche Einstellungen für das TDI-Modul-System werden in einem Accessory vorgenommen.*

Modula-Standard geben, für die ich allerdings bis heute keine Update-Benachrichtigung bekommen habe.

Das System kostet cirka DM 300 und wird auf zwei einseitigen Disketten mit Handbuch geliefert.

Die Oberfläche des Systems stellt die Text- und Codedateien als Icons dar (Bild 1). Beim Anklicken eines dieser Symbole wird je nach Dateiart der Editor, Compiler oder Linker aufgerufen. Die Bedienung ist sehr bequem und zweckmäßig. Im Vergleich zur Megamax-Shell fehlen jedoch Kommandos zur Dateiverwaltung.

Sämtliche Einstellungen für das System werden in einem Accessory vorgenommen (Bild 2). Die Suchpfade gelten dabei für alle Systemprogramme, also für die Oberfläche, den Compiler, Linker und Debugger.

Der mitgelieferte Editor ist gewöhnungsbedürftig. Er befindet sich ständig im Einfügemodus und wird sicher keine Geschwindigkeitsrekorde aufstellen. Nach längerer Arbeit damit wird man zum Verschieben von Textblöcken oder zum Vervielfältigen von Zeilen die Kombination Löschen-UNDO effizient einsetzen.

Sehr gut verarbeitet der Editor beim Übersetzen entstandene Fehlermeldungen. Kommt der Cursor an eine Fehlerstelle, erscheint die Meldung als Klartext in der Info-Zeile des Editor-Fensters.

Der Compiler arbeitet mit seinen fünf Pässen bedächtig und wird von den anderen Ein-Pass-Übersetzern übertroffen. Der erzeugte Code ist recht gut, könnte aber noch einige Optimierungen vertragen.

In Tabelle 1 finden Sie die Benchmark-Ergebnisse aller Systeme. TDI wird - abgesehen von Megamax - von allen neueren Compilern übertroffen.

Die mitgelieferten Bibliotheken sind bei den Überarbeitungen nahezu unverändert geblieben. Es sind neben den Standardmodulen Bibliotheken für AES, VDI, GEMDOS und den systemnahen BIOS-Bereich vorhanden. Module für Textfenster oder die Eventverwaltung müssen mit dem "Toolkit" extra hinzugekauft werden. Im Vergleich mit Megamax oder SPC sind die Libraries zu mager geraten.

| Modula-2 Benchmarks | | | | | | |
|---|---|---|---|---|---|---|
| Nr. | TDI V3.0 | Jefferson | Megamax | Softwave | SPC M2 | testet ... |
| 1 | 0:07 | 0:07 | 0:07 | 0:04 | 0:02 | Prozeduraufruf |
| 2 | 1:42 | 1:33 | 2:59 | 1:35 | 1:27 | Addition |
| 3 | 1:21 | 1:18 | 1:58 | 1:20 | 1:12 | Increment |
| 4 | 1:47 | 1:38 | 2:59 | 1:40 | 1:32 | Additionsoptimierung |
| 5 | 1:27 | 1:23 | 2:08 | 1:25 | 1:17 | Increment als Vergleich |
| 6 | 2:09 | 1:57 | 3:48 | 2:01 | 1:51 | INTEGER-Addition |
| 7 | 2:09 | 1:57 | 3:48 | 2:01 | 1:51 | CARDINAL-Addition |
| 8 | 1:11 | 1:18 | 1:42 | 1:05 | 1:12 | FOR-Schleife |
| 9 | 1:21 | 1:02 | 1:42 | 1:05 | 0:56 | REPEAT-Schleife |
| 10 | 1:21 | 1:18 | 1:57 | 1:20 | 1:12 | WHILE-Schleife |
| 11 | 1:04 | 0:54 | 1:15 | 0:38 | 0:48 | INTEGER-Parameter |
| 12 | 1:04 | 0:54 | 1:17 | 0:38 | 0:48 | INTEGER VAR-Parameter |
| 13 | 1:06 | 0:59 | 2:19 | 0:33 | 0:53 | RECORD-Parameter |
| 14 | 0:34 | 0:30 | 0:41 | 0:20 | 0:24 | RECORD VAR-Parameter |
| 15 | 0:49 | 0:49 | 1:33 | 0:42 | 0:43 | Konstanten-Optimierung |
| 16 | 0:51 | 0:51 | 1:33 | 0:44 | 0:45 | Konstanten-Optimierung |
| 17 | 1:28 | 1:26 | 2:06 | 1:19 | 1:20 | Expression-Optimierung |
| 18 | 1:42 | 1:22 | 1:59 | 1:15 | 1:16 | Expression-Optimierung |
| 19 | 0:37 | 0:36 | 0:55 | 0:30 | 0:30 | Zwischenergebnis-Optimierung |
| 20 | 0:37 | 0:35 | 0:55 | 0:30 | 0:30 | Zwischenergebnis-Optimierung |
| 21 | 0:09 | 0:11 | 0:13 | 0:09 | 0:05 | IF-Statement |
| 22 | 0:13 | 0:13 | 0:16 | 0:11 | 0:07 | IF durch CASE ausgedrückt |
| 23 | 0:38 | 0:33 | 0:41 | 0:28 | 0:27 | CASE-Statement |
| 24 | 0:40 | 0:39 | 1:03 | 0:38 | 0:33 | CASE durch IF ausgedrückt |
| 25 | 0:47 | 1:03 | — | 2:09 | 0:45 | REAL-Arithmetik |
| 26 | 2:05 | — | 1:32 | 2:18 | 2:07 | LONGREAL-Arithmetik |
| 27 | 1:52 | 5:42 | — | — | 3:36 | REAL-Library |
| 27a | 5:39 | — | 2:35 | 35:40 | 3:30 | LONGREAL-Library |
| 28 | 1:21 | 1:21 | 0:40 | 0:30 | 1:40 | String-Library |
| 29 | 2:10 | 2:07 | 2:13 | 1:48 | 1:40 | ARRAY-Zugriffe |
| 30 | 0:09 | 0:10 | 0:17 | 0:19 | 0:04 | RECORD-Zugriffe |
| Alle Zeiten mit time-Kommando von Gulâm gemessen<br>SPC Modula mit 200Hz-Zähler gemessen | | | | | | |

*Tab. 1: Alle Benchmarks auf einen Blick*

Der Linker arbeitet auf Wunsch optimierend und entfernt so nicht benutzte Prozeduren. Die endgültige Programmgröße ist sehr gering; kein Modula-Linker arbeitet so effizient.

Ein Debugger ist im eigentlichen System nicht enthalten. Mit dem Toolkit (s.u.) bietet TDI jedoch einen Post-Mortem-Debugger an. Er arbeitet mit einem Speicherdump nach einem Programmabsturz und kann alle Datenobjekte und den Sourcecode anzeigen. Das Programm ist ein direkter Abkömmling des ETH-Debuggers und könnte an einigen Stellen verbessert werden. Gegenüber den Möglichkeiten, die Megamax bietet, schränkt das Post-Mortem-Konzept natürlich die Anwendungsbreite ein.

Das Handbuch ist nur in englische Sprache verfügbar, dafür aber als einziges mit einer praktischen Spiralbindung versehen. Alle Fragen werden geklärt, es hat sich auch als Nachschlagwerk bewährt.

Als Ergänzung zu dem reinen Compiler-System wird der "Toolkit" angeboten. Er enthält eine Reihe von Modulen, mit denen die GEM-Programmierung erheblich vereinfacht wird. Dies reicht von der Eventverwaltung bis zu Textfenstern.

Weiterhin vorhanden ist ein Debugger, der nach einem Programmfehler eingesetzt wird und ein Post-Mortem-Debug-

ging auf Source-Code-Ebene bietet. Schließlich enthält der Toolkit das Resource-Construction-Set MMRCP.
Der große Vorteil des TDI-Systems liegt in seiner langen Marktpräsenz. Fehler sind in der dritten Version wohl nicht mehr vorhanden. Der optimierende Linker erzeugt überaus kurze Programme und ist nach wie vor der beste Modula-Linker für den ST.
Im Vergleich mit den anderen Modula-Paketen schneidet TDI bei der Übersetzungsgeschwindigkeit, beim erzeugten Code und im Lieferumfang schlecht ab. Erst mit dem Toolkit, der extra gekauft werden muß, kann sich das System ernsthaft mit der Konkurrenz messen.

## Jefferson Modula

Jefferson-Modula ist bisher nur als Import erhältlich. Wegen seines geringen Preises von $50 handelt es sich dabei um einem Preisbrecher. Es ist das einzige System unter DM 100 und wird auf einer einseitigen Diskette geliefert.
Als Umgebung wird bei Jefferson die leistungsstarke Public-Domain-Shell Guläm verwendet (Bild 3). Wer auf GEM-Umgebung verzichten kann und will erhält eine erstklassige Kommandoshell, die bei richtiger Benutzung eine große Unterstützung ist. Im Vergleich mit den aufwendigen GEM-Shells anderer Systeme wirkt diese Ausstattung allerdings etwas altertümlich.
Ein extra Editor wird nicht mitgeliefert. Das ist allerdings auch nicht notwendig, da der Guläm einen EMACS-Editor eingebaut hat. Ein wirkliches Problem ist dagegen die amerikanische Tastaturanpassung, die für STs mit German-Keyboard absolut unverwendbar ist. Guläm läßt sich allerdings entsprechend patchen - Anpassungsarbeiten, die man bei einem Import in Kauf nehmen muß.
Der 1-Pass-Compiler arbeitet wie der Linker eher schweigsam (Bild 4). Er liefert einen überraschend guten Code, der die erheblich teureren Systeme TDI und Megamax abhängt.
In den Benchmarks macht Jefferson eigentlich eine gut Figur. Alle Ergebnisse lassen eine gute Optimierung erkennen und kommen den Spitzenreitern SPC und Softwave nahe.
Allerdings implementiert Jefferson Modula-2 nicht komplett: Die Coroutinen sind dem Compiler nicht bekannt und werden auch nicht durch ein Modul zugänglich gemacht. Ebenso fehlen die LONGREALs, so daß der Sprachumfang doch merklich eingeschränkt ist.

```
yet another shell for  Atari ST

(non-profit redistribution is permitted)

»guläm«  Copyright © 1987 pm@Case

61 built-in commands:
alias    dirc     exit     history  msoff    pwd      source   unalias
cd       dirs     fg       if       mv       rehash   sx       unset
chmod    echo     fgrep    lpr      peekw    ren      te       unsetenv
copy     egrep    foreach  ls       pokew    rm       teexit   which
cp       ef       format   mem      popd     rmdir    time     while
date     endfor   gem      mkdir    print    rx       touch
df       endif    grep     more     printenv set      ue
dm       endwhile help     mson     pushd    setenv   uekb

programs accessible through hash table:
oops   m2link m2dec  m2
d:\jm2 58#
d:\jm2 58# ls bin
bin

d:\jm2\bin:
m2.ttp        m2dec.ttp  m2link.ttp oops.tos
d:\jm2 59#
```

*Bild 3: Jefferson-Modula benutzt die Guläm-Shell*

Die mitgelieferten Bibliotheken sind nicht sehr umfangreich. Es sind die Standard-Module vorhanden; sie werden ergänzt durch die normalen Bibliotheken für AES, VDI, GEMDOS und BIOS. Module für höhere GEM-Funktionen sind nicht vorhanden und müssen selber geschrieben werden.
Fehlerfrei sind die Module sicher nicht; Versuche, das InOut-Modul zu benutzen, führten zu einem Systemabsturz. Durch Fehler und Ungereimtheiten macht die Auslieferung einen etwas schlampigen Eindruck.
Der Linker arbeitet zufriedenstellend und liefert nicht auf Größe optimierte Programme.
Nennenswerte Debugging-Hilfen gibt es nicht. Im Gegenteil: Um aus den Compiler-Fehlermeldungen Klartext zu erzeugen, ist der Lauf eines kleinen Programms namens OOPS notwendig, was die Programmentwicklung unnötig erschwert.
Das Handbuch ist in der Version für $50 ein Heftchen mit 26 Seiten. Ein erweitertes Handbuch muß zusätzlich gekauft werden. Für die Installation des Systems reicht es aus, geht allerdings kaum ins Detail. Gegenüber den ausführlichen

```
d:\jm2 64# m2 allqueen.mod
Jefferson Software Modula-2 Compiler V-1.00

allqueen.mod
 - d:\jm2\std\InOut.SBM
 + AllQueens.RFM
 + AllQueens.OBM 584
d:\jm2 65#
d:\jm2 65# m2link allqueen
Jefferson Software Linker V-1.00

 - 0000003C allqueen.OBM
 - 000002EE d:\jm2\std\InOut.OBM
 - 00001246 d:\jm2\std\FileSystem.OBM
 - 00001EE8 d:\jm2\std\System.OBM
 - 000028DC d:\jm2\tos\GEMDOSFiles.OBM
 - 00002D26 d:\jm2\std\FPSys.OBM
 + 00003562 allqueen.TOS
d:\jm2 66# ls -l allqueen.*
-rw------- 1 u            783 Feb 13 19:34 allqueen.mod
-rw------- 1 u            720 Feb 18 15:38 allqueen.obm
-rw------- 1 u            357 Feb 18 15:38 allqueen.rfm
-rw------- 1 u          13881 Feb 18 15:39 allqueen.tos
d:\jm2 67#
```

*Bild 4: Compiler und Linker des Jefferson-Modula bei der Arbeit*

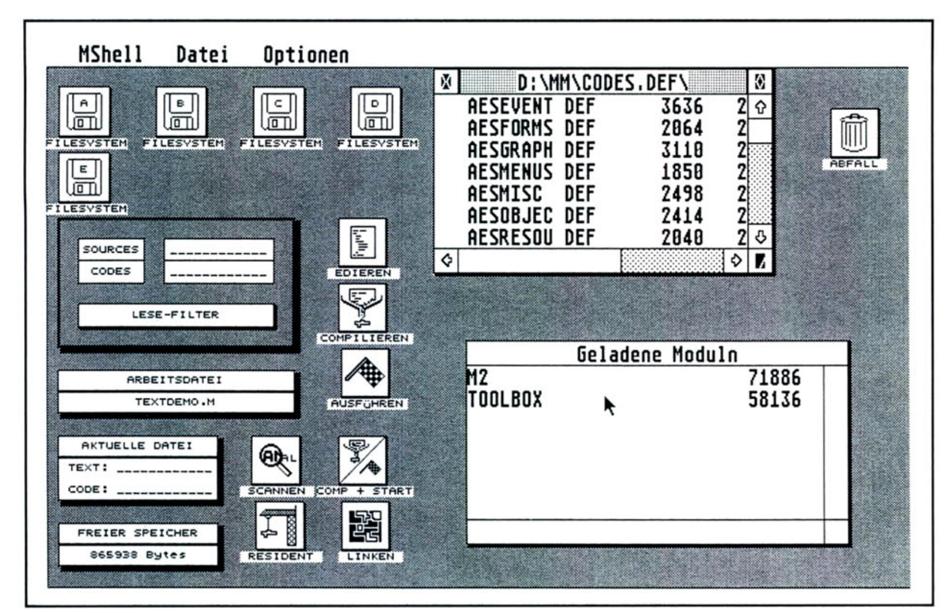



*Bild 5: Megamax Modula bietet die beste Shell.*

Anleitungen der anderen Systeme ist Jefferson-Modula sehr schlecht dokumentiert.

Jefferson-Modula ist das preisgünstigste Produkt für den ST. Ob der Niedrigpreis wirklich die Abstriche bei der Ausstattung rechtfertigt, ist zweifelhaft. Auch durch das Fehlen eines bundesrepublikanischen Vertriebs fällt Jefferson als wirkliche Alternative aus.

## Megamax Modula

Megamax-Modula-2 war sehr lange angekündigt und ist in diesem Frühjahr endlich erschienen. Für DM 398,- erhält man vier teils doppelseitige Disketten und ein Handbuch im Ringordner.

Die Umgebung für Megamax-Modula ist mit über 200 KiloByte die größte, aufwendigste und leistungsstärkste unter den getesteten Paketen (Bild 5). Das Programm stellt praktisch alle Funktionen des GEM-Desktops zur Verfügung, so daß man die Shell nicht zum Kopieren oder Löschen von Dateien verlassen muß.

Alle Funktionen werden über die Icons aufgerufen, wobei es die meisten Aktionen auch als Tastenkombinationen gibt. Um Ladezeiten zu sparen, lassen sich Module oder Programm auch resident im Speicher halten. In der abgebildeten Situation sind der Compiler und der Editor geladen, so daß der Turn-Around-Edit-Compile ohne Massenspeicherzugriff abläuft.

Mit einer umfangreichen Installationsdatei läßt sich das System sehr flexibel an die Ordnerstrukturen anpassen. Hier lassen sich auch Voreinstellungen setzen oder ein anderer Editor problemlos einbinden.

Die Shell übernimmt noch ein Konzept, das die Programmentwicklung beschleunigen soll: das Load-Time-Linking. Dabei müssen Programme zum Starten nicht mehr durch einen Linker geschickt werden, vielmehr bindet die Shell beim Ladevorgang alle benötigten Module zusammen. Ergebnis ist eine schnellere Programmentwicklung, da kein extra Linkerlauf notwendig ist.

Als Editor liegt dem System eine Version des Toolbox-Editors bei, den Application Systems auch einzeln als C-Quelle anbietet. Er ist nicht besonders auf Modula zugeschnitten, fügt sich allerdings recht gut in das System ein.

Der Compiler kann direkt vom Editor aufgerufen werden und auch die Anzeige von Fehlerstellen geht problemlos. Das Programm funktioniert ansprechend schnell und erfüllt seine Aufgabe. TEMPUS-Fans können ihren Editor problemlos einbinden.

Der Compiler ist die Schwachstelle des Systems. Zwar kann er dank Assemblerprogrammierung und 1-Pass-Konzept mit rasender Geschwindigkeit arbeiten, der erzeugte Code erwies sich aber als langsam und kaum optimiert. Lediglich die LONGREAL-Arithmetik kann überzeugen.

In der Benchmark-Tabelle ist Megamax die Schlußleuchte. Die Zeiten liegen weit hinter denen der anderen Systeme. In einzelnen Fällen wird glatt die doppelte Rechenzeit benötigt.

So komfortabel die Entwicklungsumgebung auch sein mag, das eigentliche Ergebnis der Programmierarbeit läßt zu wünschen übrig. Application Systems hat an dieser Stelle Besserung versprochen; die Optimierung lag für diesen Artikel jedoch noch nicht vor.

Für zeitkritische Anwendungen wurde die Sprache um einen Assembler erweitert. In einem ASSEMBLER-END-Block können alle 68000-Mnemoniks verwendet werden, wobei der Zugriff auf Modula-Variablen problemlos möglich ist.

Megamax liegen die umfangreichsten Bibliotheken der getesteten Systeme bei. Sie gehen weit über die "normalen" Libraries mit AES/VDI sowie GEMDOS/BIOS-Zugriffen hinaus.

Funktionen für eine vereinfachte AES-Programmierung erlauben Textfenster, eine vereinfachte Eventprogrammierung und komfortablen Zugriff auf Objektbäume.

Weiter finden sich sehr viele Routinen, die das Programmieren erheblich beschleunigen. Zugriff auf Listenstrukturen, komfortable Konvertierungroutinen für Zahlen und Strings sind nur ein paar der vielen mitgelieferten Module.

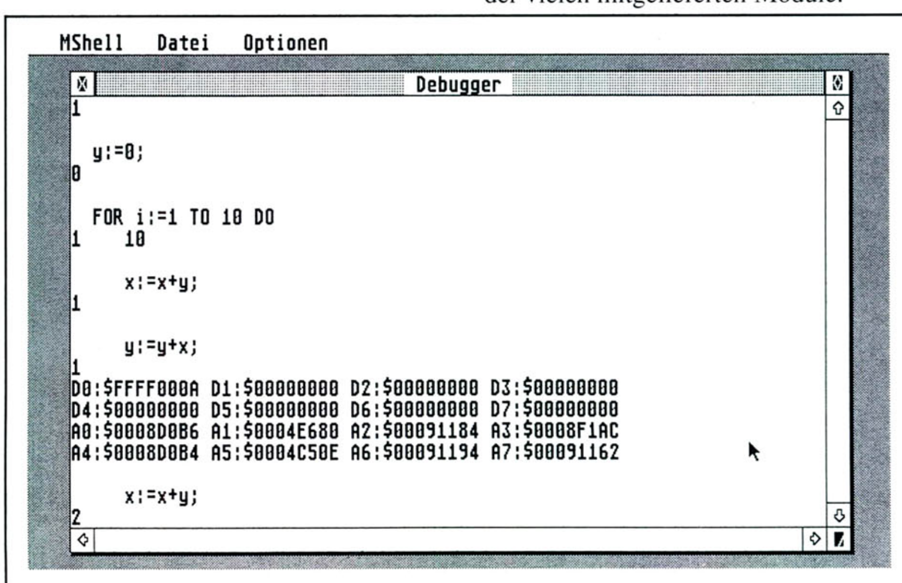



*Bild 6: Single-Steps mit Megamax Modula*

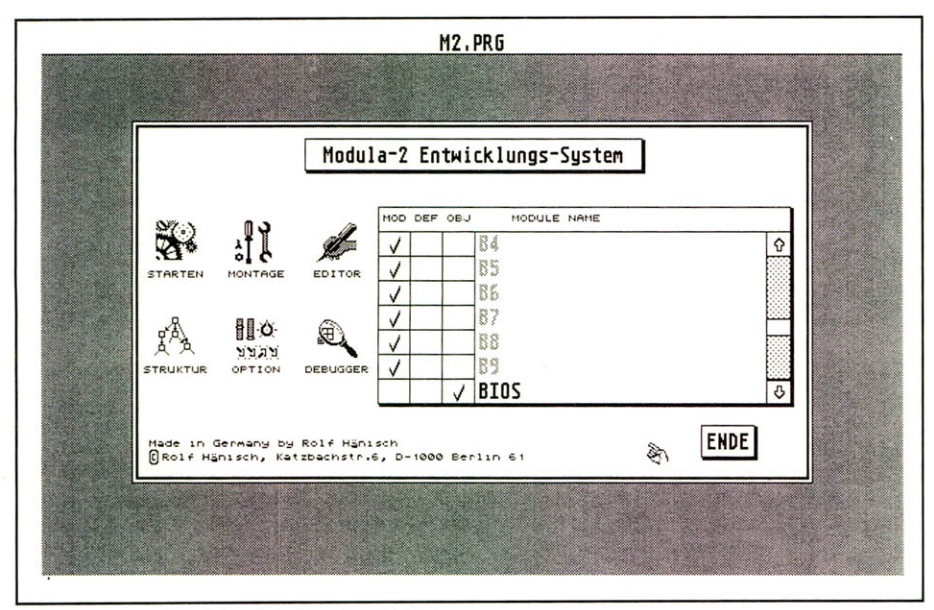



*Bild 7: Softwave Modula wird über eine Dialogbox gesteuert*

Auch wenn die Shell ein Load-Time-Linking erlaubt, liegt für Stand-Alone-Programme ein separater Linker bei. Er arbeitet nicht optimierend und erzeugt somit Programme, die größer als notwendig sind. Ein optimierender Linker wird ab Anfang Juli erhältlich sein.

Das Debugging gestaltet sich bei Megamax sehr komfortabel. Bei einem Laufzeitfehler erscheint zunächst eine Alertbox, die einige Informationen über die Absturzstelle ausgibt und eine Inspektion der Ausrufkette ermöglicht.

In einer zweiten Alertbox kann man dann entscheiden, ob der Fehler ignoriert, das Programm abgebrochen oder der Editor gestartet werden soll. Der Cursor wird dann im Quelltext auf die fehlerhafte Stelle gesetzt.

Aber auch schon während des Programmlaufs ist eine Überwachung möglich. Das Programm läuft dann in einem Single-Step-Modus und gibt in einem Fenster das Ergebnis des jeweils berechneten Ausdrucks aus (Bild 6). Der Debugger erlaubt einige Analysen über Registerinhalte oder Codeadressen. Die Debug-Optionen können auch über Steuervariablen des Debug-Moduls vom Programm selber verändert werden.

Das Handbuch ist sehr gut gelungen. Den größten Teil nehmen, wie sonst auch, die Listings der Definitions-Module der Bibliotheken ein. Dabei werden die Routinen in Kommentaren so gut dokumentiert, daß kaum Fragen offen bleiben. Der erste Teil erläutert die Systembenutzung und beschreibt den Compiler. Dabei bekommt der Anfänger eine leicht verständliche Anleitung und der Experte alle Fakten, die notwendig sind, um z.B. mit dem Assembler auf Modula-Objekte zuzugreifen. Ein sehr gutes Handbuch mit umfangreichem Register, das für andere Systeme Vorbild sein sollte.

Megamax liegt ein Resource-Construction-Set standardmäßig bei. Es handelt sich um das NRSC von Kuma in einer deutschen Version, die auch unter Blitter-TOS läuft. Falls das eingebaute Debugging nicht ausreicht, kann man auch den mitgelieferten TEMPLMON-Monitor verwenden, der wohl die allerletzten Fehler aufzeigt.

Megamax unterstützt die Programmentwicklung hervorragend. Shell und Bibliotheken sind die Glanzstücke des Pakets. Abstriche müssen allerdings beim erzeugten Code gemacht werden.

## Softwave Modula

Das mit DM 199,- preiswerteste, direkt erhältliche System wird auf einer Diskette geliefert. Dazu kommt ein Handbuch im Ringordner.

Das System integriert alle Bestandteile in einem Programm. Das integrierte Paket wird über eine Dialogbox (Bild 7) gesteuert. Beim Festlegen der Ordnerstrukturen werden alle Dateien auf Abhängigkeiten hin untersucht. Die Auswahl einer Arbeitsdatei per Maus in der Fileliste bringt dann alle abhängigen Module in normaler Schrift auf den Bildschirm.

Die Shell hat ein Make eingebaut, bei dem aufgrund der festgestellten Abhängigkeiten alle zusammenhängenden Module automatisch neu compiliert werden, falls notwendig.

Der Editor ist sehr einfach programmiert und verfügt noch nicht über eine Menüsteuerung. Als Sprachunterstützung sind Textmakros möglich, mit denen Gerüste für oft verwendete Strukturen oder Schlüsselworte automatisch erzeugt werden können.

Der Compiler erzeugt recht gut optimierten Code. Schwächen haben allerdings noch die Fließkommaoperationen, bei denen das System das Schlußlicht bildet. Nach den Benchmarks hat es zusammen mit SPC die beste Codegenerierung.

Nach und nach soll der Übersetzer auch an andere Prozessoren der Motorola-Familie angepaßt werden, so daß er eventuelle Zusatzprozessoren unterstützt (Bild 8).

Einige kleinere Spracherweiterungen implementieren einen STRING-Typ und zusätzliche eingebaute Funktionen. Allerdings sind diese Features in der aktuellen Version noch nicht sehr nützlich.

Die Bibliotheken enthalten wie bei TDI nur die Standardmodule und die normalen AES/VDI- und GEMDOS/BIOS-Libraries.

Unter dem Montage-Icon verbirgt sich der Linker. Zunächst löst das Make not-

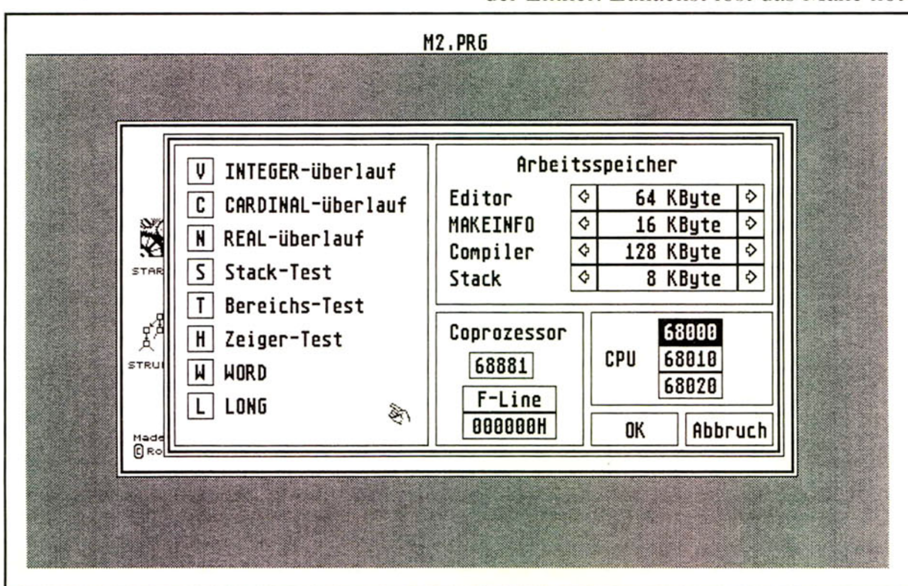



*Bild 8: Softwave Modula soll auch an andere Motorola-Prozessoren angepaßt werden.*

wendige Übersetzungen aus, worauf der Linker dann die Module zusammenbindet. Er arbeitet nicht optimierend.
Besondere Debugging-Möglichkeiten gibt es nicht. Es ist allerdings ein Debugger auf Source-Code-Ebene angekündigt. Zum Zeitpunkt dieses Vergleichs lag er jedoch noch nicht vor.
Das Handbuch erläutert ausreichend die Benutzung des Systems. Ein längeres Kapitel soll in die Modula-Programmierung einführen, und abschließend werden wie üblich die Definitionsmodule der Libraries aufgelistet. Dabei sollte etwas mehr an erläuternden Kommentaren geboten werden. Die Druckvorlage für das Handbuch ist mit einem Matrixdrucker erstellt - vielleicht sollte der Hersteller hier einen Laserdrucker verwenden.
Softwave-Modula ist ein System, das sehr gut optimierten Code liefert. Die Systemintegration und das Make vereinfachen die Programmentwicklung. Schwachpunkte sind die nicht sehr umfangreichen Bibliotheken, die Real-Arithmetik und der noch nicht komplette Editor. Durch den Preis bietet das System aber einen kostengünstigen Einstieg in Modula-2.

## SPC Modula

SPC-Modula ist für DM 349,- auf einer Diskette erhältlich. Das System ist eine direkte Portierung des ETH-Compilers.
Als Umgebung wird eine tastenorientierte Shell mitgeliefert (Bild 9). Sämtliche Systemprogramme werden mit einem Buchstaben aufgerufen, wobei die Shell die wahrscheinlich nächste Aktion als Default anbietet. Wenn sich die Softwareentwicklung im herkömmlichen Edit-Compile-Test-Zyklus vollzieht, können alle Aufrufe durch mehrmaliges Drücken von <Return> durchgeführt werden.
Der Shell fehlen dafür jegliche Dateioperationen und der grafische Komfort, den vor allem Megamax bietet.
Der mitgelieferte Editor unterstützt die Entwicklung von Modula-Programmen besonders. Die "Language-Support"-Option bewirkt, daß alle in Kleinbuchstaben eingegebenen Wörter möglichst früh in Modula-Schlüsselworte umgewandelt werden. Um "PROCEDURE" einzugeben, reichen dann schon die Buchstaben "pr".
Ebenfalls strukturunterstützend lassen sich die Textmakros verwenden, die auf den Zehnerblock gelegt werden.
Der Compiler arbeitet schnell und liefert gut optimierten Code. Seine Benchmarks

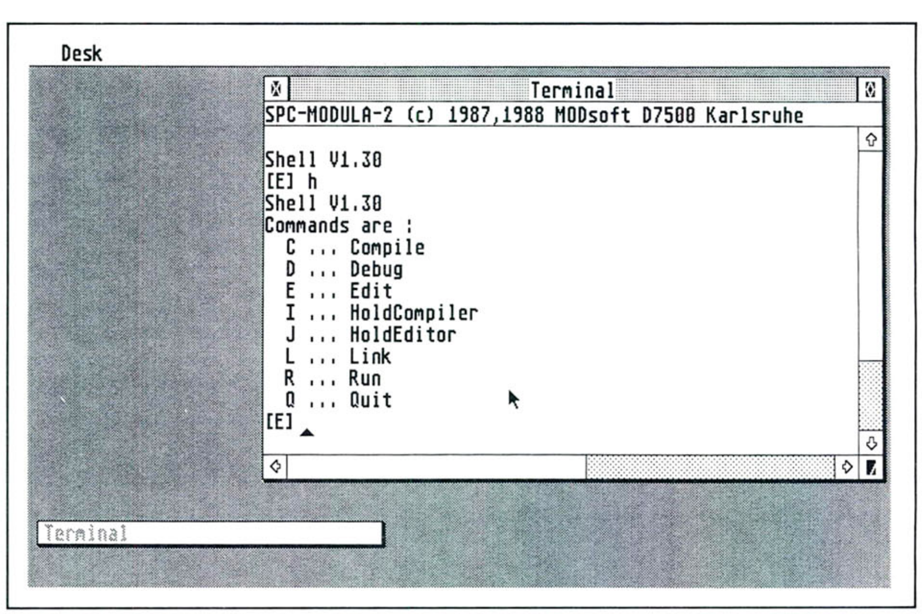


*Bild 9: SPC Modula arbeitet mit einer tastenorientierten Shell.*

werden nur noch vom Softwave-System übertroffen. Dabei sind die REALs mit einfacher Genauigkeit etwas langsam, da sie vor der Berechnung in LONGREALs umgewandelt werden.
Die gemessenen Zeiten sind mit dem Softwave-System vergleichbar und machen SPC zu einem sehr guten Produkt, soweit es um die Codegeschwindigkeit geht.
Die Bibliotheken umfassen den Standard und die gewohnten Atari-Libraries mit AES/VDI und GEMDOS/BIOS. Dazu kommen einige Hilfsmodule mit erweiterten Funktionen für Strings oder das Filesystem.
Für fensterorientierte Anwendungen existiert das "SSWiS"-Modul, das eine rechnerunabhängige Programmierung von grafischen Oberflächen ermöglicht.
Die Bibliotheken könnten eine hohe Portabilität von Programmen ermöglichen, falls SPC auch auf andere Rechner übertragen werden sollte. Damit wird natürlich ein Geschwindigkeitsverlust in Kauf genommen. Ein Fenstersystem, das eine völlig GEM-unabhängige Schnittstelle hat, verursacht natürlich mehr Rechen-Overhead als ein spezialisiertes GEM-Fenstermodul.
Ein Linker lag zum Zeitpunkt dieses Vergleichstests nicht vor. Er ist angekündigt und soll im Sommer geliefert werden. Dadurch sind im Moment keine Stand-Alone-Programme möglich (man denke nur an Accessories), und SPC hätte meiner Meinung nach vielleicht zunächst andere Bestandteile zurückstellen und das Paket von Anfang an in den Grundfunktionen komplett ausliefern sollen.
Als Debugger steht der Post-Mortem-Debugger aus dem ETH-System zur Verfügung (Bild 10). Er wird nach einem Programmabsturz eingesetzt und erlaubt dann die Inspizierung des Programmzustands einschließlich Source-Anzeige. Er entspricht über weite Teile dem TDI-Debugger.

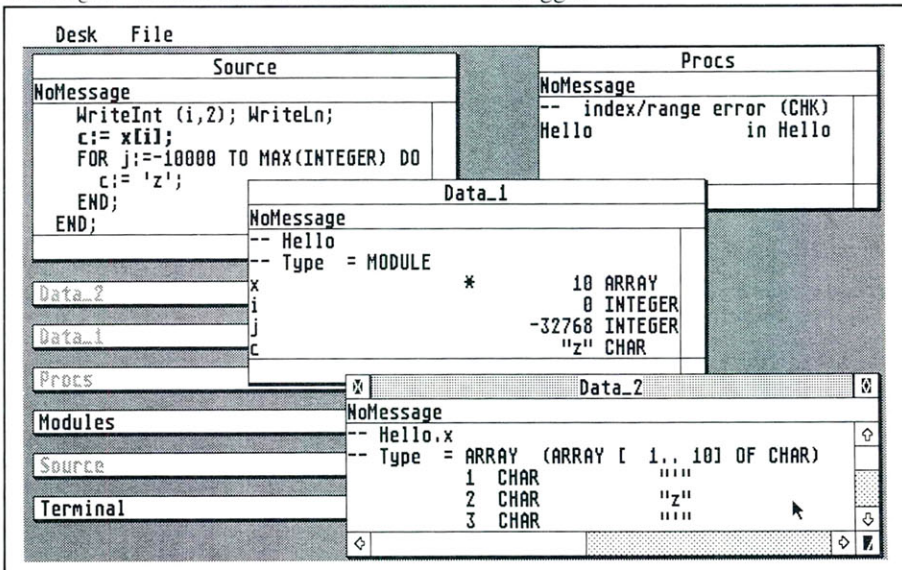


*Bild 10: Der Post-Mortem-Debugger des SPC Modula.*

# DAS TUNING-PROGRAMM FÜR IHREN ATARI ST

***Lassen Sie Ihrem 68000er freien Lauf!***

**Woran liegt es, daß manche Atari-Anwender anderen eine Nasenlänge voraus sind?**

Sie sind einfach besser informiert. Verfügen über mehr Know-how zu ihrem 68000er-Rechner. Sind immer auf dem neuesten Stand.

Ab sofort schöpfen auch Sie die Möglichkeiten Ihres 68000ers voll aus: Zwei neue Nachschlagewerke geben Ihnen praxisnahe, detaillierte Informationen.

Damit

- wissen Sie Bescheid über aktuelle Hard- und Software
- erschließen Sie sich mit selbstgebauten Systemerweiterungen und Bausteinprogrammierung neue Einsatzbereiche und
- programmieren hardwarenah in Assembler und C

**Ihr besonderer Vorteil:**
Ein eigener Update- und Ergänzungsservice garantiert Ihnen auch in Zukunft ständige Aktualität.

***Blättern Sie bitte um ...***
und informieren Sie sich auf der Rückseite, was Ihnen diese Werke im einzelnen bieten.

Diese beiden Nachschlagewerke sorgen dafür, daß Sie die Möglichkeiten Ihres Atari ST optimal nutzen:







INTEREST-VERLAG
Industriestraße 21
D-8901 Kissing
Tel. 0 82 33/21 01-0


60 Pfennig, die sich lohnen!

Postkarte/Antwort

INTEREST-VERLAG
Fachverlag für anspruchsvolle Freizeitgestaltung
Herrn Michel
Industriestraße 21

D-8901 Kissing

160061 IN

# Für Ihren 68000er

## Lassen Sie Ihrem 68000er freien Lauf!

... und schöpfen Sie die enormen Fähigkeiten Ihres 68000er-Systems optimal aus:

Dieses neue Nachschlagewerk gibt Ihnen

- detailliertes Wissen über Aufbau und Funktionen der 68000er-Prozessoren und ihrer Peripheriebausteine
- fundierte Betriebssystemkenntnisse
- das Rüstzeug zum hardwarenahen Programmieren in Assembler und C
- Tools und Utilities

**Auszüge aus dem Inhalt:**

**Die Prozessorfamilie 68000:** mechanische und elektrische Spezifikation, Registerfunktionen und Exception-Verarbeitung, Busoperationen




**Betriebssystem-Know-how** zu OS-9, GEM-TOS

**Softwareengineering:** der optimale Weg von der Idee zum Programm

**Programmierkurse** für Assembler und C

**Tools und Utilities:** u.a. universeller symbolischer Disassembler, Grafikutilities für dreidimensionale Projektionen

**68000er-Applikationen:** Schaltungs- und Programmentwicklung, von der Digitalelektronik zum CNC-System

**Systemtuning durch Hardwareerweiterungen:** u.a. Bauanleitungen für universelle 16-Kanal-E/A-Interfaces, die Ihnen Meß-Steuer-Regelanwendungen ermöglichen.

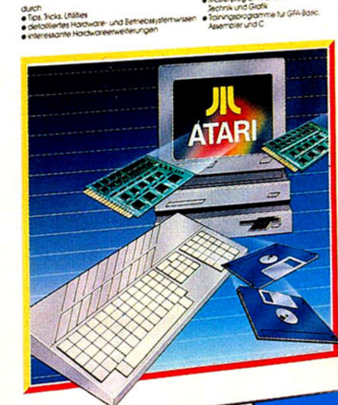
**Zusätzlicher Diskettenservice für jedes Werk**

## Hier ist das Tuning-Programm für Ihren Atari ST!

**Hardware- und Systemerweiterungen für neue Einsatzbereiche**

Bauen Sie selbst eine **Harddisk**, und Sie erzielen kürzeste Zugriffszeiten für schnelles Erfassen großer Datenmengen.

Der Ausbau der Ports, der Einsatz eines universellen Parallelinterfaces oder die Programmierung von E/A-Bausteinen ermöglichen Ihnen z.B. den Aufbau eines universellen Meß-/Steuer-/Regelsystems.




**Perfekte Programmiertechnik**
... eignen Sie sich in Kursen zu **C, 68000er-Assembler, GFA-Basic und Pascal** an. Jetzt wählen Sie die Sprache aus, mit der Sie je nach Aufgabenstellung das optimale Ergebnis erzielen.

**Raffinierte Tips, Tricks und Utilities** wie z.B. ein Programm zur **Hochformatierung der Floppy auf 946 KB** machen Ihre Programme noch schneller und besser. Dieses Werk bietet Ihnen u.a. Routinen für Bildschirm (Escape-Sequenzen, Alertbox V 1.0, Blendenroutine), Drucker (Neoprint-, Neoscan-Routinen) sowie für Grafik (Geminit Headerfile) und Sound (Digi-Sound-Routine).

**Sofort einsetzbare Musterlösungen** geben Ihnen u.a. den Einstieg in Desktop-Publishing und Sound-Sampling.

**Detailliertes Hardware- und Betriebssystem-Know-how:** Damit nutzen Sie z.B. die RS 232-Schnittstelle zur Rechnerkopplung.

Treiben Sie's ruhig bunt! Die **Assemblerbibliothek Transcolour** ermöglicht Ihnen anspruchsvolle Rastergrafik in selbstdefinierten Farbbereichen mit bis zu 512 Farben.

Bitte abtrennen

## ☒ Ja, senden Sie mir bitte sofort

(Bitte Zutreffendes ankreuzen)

☐ Expl. **Erfolgreiches Programmieren von 68000er-Systemen in Assembler und C**
Stabiler Ringbuchordner, Format DIN A4, Grundwerk über 400 Seiten, Bestell-Nr.: 3400 · Preis: DM 92,––.

Alle 2–3 Monate erhalten Sie Ergänzungsausgaben zum Grundwerk mit je ca. 120 Seiten zum Seitenpreis von 38 Pfennig (Abbestellung jederzeit möglich).

☐ Expl. **Erfolgreiches Hard- und Softwaretuning der Atari ST 260/520/1040 und Weiterentwicklungen**
Stabiler Ringbuchordner, Format DIN A4, Grundwerk ca. 450 Seiten, Bestell-Nr.: 3700 · Preis: DM 92,––.

Alle 2–3 Monate erhalten Sie Ergänzungsausgaben zum Grundwerk mit je ca. 120 Seiten zum Seitenpreis von 38 Pfennig (Abbestellung jederzeit möglich).

Meine Anschrift:

Name, Vorname

Straße, Hausnummer

PLZ, Wohnort

Datum ✗ Unterschrift

Bitte unterschreiben Sie auch Ihre Sicherheitsgarantie,

mit der Sie folgendes zur Kenntnis nehmen: Sie haben das Recht, Ihr angefordertes Werk sowie jede nachfolgende Ergänzungsausgabe innerhalb von 10 Tagen an den INTEREST-VERLAG, Industriestr. 21, 8901 Kissing zurückzusenden, wobei zur Fristwahrung das Datum der Absendung genügt. Sie kommen dadurch von allen Verpflichtungen aus der Bestellung frei.

Datum ✗ Unterschrift





**Erst prüfen, dann kaufen:**

Sehen Sie sich jedes Werk in Ruhe zu Hause an. 10 Tage lang dürfen Sie Ihr Ansichtsexemplar prüfen.

**Fordern Sie noch heute an:**

**Erfolgreiches Programmieren von 68000er-Systemen in Assembler und C**
stabiler Ringbuchordner, DIN A4, Grundwerk über 400 Seiten, Bestell-Nr.: 3400, Preis: DM 92,–

**Erfolgreiches Hard- und Softwaretuning der Atari ST 260/520/1040 und Weiterentwicklungen**
stabiler Ringbuchordner, DIN A4, Grundwerk ca. 450 Seiten, Bestell-Nr.: 3700, Preis: DM 92,–

| | TDI V3.0 | Jefferson | Megamax | Softwave | SPC-Modula |
|---|---|---|---|---|---|
| Compiler-Pässe | 5 | 1 | 1 | 2 | 1 |
| Geschwindigkeit lt. Hersteller | k.A. | k.A. | 5000 | 12000 | 5000 |
| INTEGER/CARDINAL/Bits | 16 | 16 | 16 | 32 | 16 |
| LONGINT/LONGCARD/Bits | 32 | 32 | 32 | 32 | 32 |
| REAL/Bits | 32 | 32 | 64 | 32 | 32 |
| LONGREAL/Bits | 64 | 64 | 64 | 64 | 64 |
| SETS/maximale Elemente | 65 536 | 16 | 65 536 | 32 | 16 |
| BITSET/maximale Elemente | 16/32 | 16 | 16 | 16 | 16 |
| Zusätzliche Bibliotheken | nur im Toolkit | nein | ja | nein | ja |
| Resource-Construction-Set | nur im Toolkit | nein | ja | nein | nein |
| Systemgröße/KiloByte | 620 | 760 | 1680 | 350 | 520 |
| Handbuchstärke/Seiten | 370 | 26 | 380 | 160 | 210 |
| Preis | 298,- | $50 | 398,- | 199,- | 349,- |

*Tabelle 2: Weitere Daten im Überblick*

Das Handbuch erläutert das System brauchbar. Ein gesonderter Teil soll in die Modula-Pogrammierung einführen. Dabei sind zum Zeitpunkt dieses Vergleichs einzelne Kapitel noch als "wird nachgereicht" angekündigt.

Die Dokumentation der mitgelieferten Bibliotheken enthält leider zu wenig Kommentare, um alle Fragen zu klären. So wird z.B. das Modul "Process", das die Coroutinen auf einem logisch höheren Level anbietet, so unzureichend beschrieben, daß eine praktische Arbeit damit nicht möglich ist.

SPC-Modula hat gegenüber den Konkurrenten aus der Spitzengruppe bei der Codegeschwindigkeit die besseren Bibliotheken zu bieten. Megamax, das in dieser Hinsicht noch besser ist, kann bei der Codeerzeugung nicht mithalten.

SPC ist ein gutes System, bei dem zwar noch nicht alles optimal ist, das aber gute Leistungen in allen Aspekten bietet.

## Das Traumsystem

Da sich kein System als in allen Bestandteilen optimal bezeichnen läßt, möchte ich noch kurz das ultimative Modula-System für den ST skizzieren. Zusammenkommen müßte eine Oberfläche wie bei Megamax, die erheblich mehr kann, als nur einen komfortablen Wechsel zwischen den Paket-Bestandteilen zu ermöglichen. Dazu käme ein Compiler, der schnell läuft und einen Code wie Softwave oder SPC liefert.

Der beste Linker ist im Moment bei TDI zu finden, bei den Bibliotheken hat Megamax das meiste zu bieten, ebenso wie beim Debugging und dem Handbuch.

Ein Resource-Construction-Set wäre auch unabdinglich. Und um diesen kleinen Ausflug ins Reich der Phantasie zu vervollständigen käme ein Preis wie bei Softwave oder gar Jefferson in Frage.

In den Tabellen 2 und 3 finden Sie weitere technische Daten der Systeme und nochmals alle Stärken und Schwächen auf einen Blick.

## Anmerkungen

Nachdem nun eine Reihe von Compilern vorliegt, möchte ich einige allgemeine Bemerkungen zu den Modula-Systemen für den ST machen.

Man möchte meinen, daß Modula-Programme portabel sind, wenn systemabhängige Funktionen in eigenen Modulen versteckt werden. Doch kaum ein Programm läßt sich unter den beschriebenen Compilern ohne Überarbeitung austauschen. Selbst die kleinen Benchmark-Programme, die von vornherein als reinstes Standard-Modula formuliert wurden, müssen immer wieder angepaßt werden.

Dafür gibt es mehrere Gründe. Da sind zunächst die unterschiedlichen Größen von INTEGER, CARDINAL und REAL. Wirth hat keinerlei Aussagen darüber gemacht, ob nun ein INTEGER zwei oder vier oder gar acht Bytes belegen soll. Dafür gibt es z.B. die unterschiedlichen Typen SHORTINT, INTEGER und LONGINT, die eine gewisse Abstufung erlauben.

Bei den Modula-Implementierungen legen sich einige Entwickler auf eine Länge fest, was zur Folge hat, daß z.B. INTEGER völlig identisch ist mit LONGINT. Doch ein Programmierer kann verlangen, daß er mittels der Typangabe auch bestimmen kann, welche Objektgröße verwendet werden soll. Wenn man als Grundtyp ein 32-Bit-INTEGER benutzt, sollte zumindest auch ein SHORTINT vorhanden sein, das 16 Bit groß ist.

Ähnlich ist die Lage bei den REAL-Implementierungen. Wenn hier REAL immer mit doppelter Genauigkeit berechnet wird, ist der Geschwindigkeitsnachteil offensichtlich. Man kann verlangen, daß ein Modula-Compiler alle denkbaren Grundtypen auch wirklich implementiert.

Die Bibliotheken sind der Dreh- und Angelpunkt von Modula-Programmen. Daß Wirths Vorschlag für eine Standardbibliothek nicht ausreicht, ist völlig klar. Doch immerhin wurde damit ein minimaler Standard festgesetzt, und es ist unverständlich, warum niemand ihn einhalten kann. Auch wenn ein System bessere Bibliotheken für Standardaufgaben enthält, würde ich erwarten, daß wegen der Portabilität auch die rudimentären Wirth-Module vorhanden sind.

Eine weitere Unsitte bei den Modula-Compiler auf dem ST sind eigene Spracherweiterungen. Das Hinzufügen eines STRING-Typs z.B. ist nicht einfach ein "Feature" des Systems, es ist vielmehr eine grundlegende Änderung der Sprache an sich. Das gleiche gilt für einen eingebauten Assembler.

Erweiterungen gehören in eigene Module. Wenn sich Routinen nur schlecht als Prozeduren formulieren lassen, z.B. wegen wechselnder Parameteranzahl, muß man eben Modula-Tricks anwenden oder das Konzept überdenken. Ein Wildwuchs von Designänderungen kann der Sprache nur schaden und nutzt das Modul-Konzept nicht aus.

Bei umfangreichen Bibliotheken wächst der Verwaltungsaufwand auf den Massenspeichern enorm an. Hat man 50 Module, so entstehen bei getrennten Symbol- und Linkdateien schon 100 Files, die im Zugriff sein müssen. Schon jetzt ist die Arbeit mit nur einem Diskettenlaufwerk praktisch unmöglich, richtig komfortabel wird's erst mit Festplatte.

Vielleicht sollten Implementierungen auf dem ST ein gepacktes Libraryformat verwenden, wie es auch auf Großrechnern zu finden ist. Alle Standardmodule gepackt in eine Datei STANDARD.LIB mit einigen simplen Zugriffsmechanismen für Compiler und Linker, und schon wäre Benutzern mit Diskettenlaufwerken erheblich geholfen.

Bis jetzt gibt es praktisch keine Tools für die Modula-Systeme. Zwar ist ab und zu

| | |
|---|---|
| TDI V3.0 | ⊕ Mehrmals überarbeitetes System<br>⊕ Optimierender Linker<br>⊖ Zusätzliche Ausgaben für Toolkit |
| Jefferson | ⊕ Niedrigpreis<br>⊖ Keine Coroutinen<br>⊖ Wenig Bibliotheken<br>⊖ Schlampige Auslieferung<br>⊖ Muß in USA gekauft werden |
| Megamax | ⊕ Umfangreiche Bibliotheken<br>⊕ Komfortable Oberfläche<br>⊕ Sehr gutes Handbuch<br>⊖ Langsamer Code |
| Softwave | ⊕ Optimierender Compiler<br>⊕ Integriertes System<br>⊕ eingebautes Make<br>⊖ wenig Bibliotheken |
| SPC-Modula | ⊕ Hochoptimierender Compiler<br>⊕ überzeugende Oberfläche<br>⊕ Zusätzliche Bibliotheken |

ein Make oder Cross-Referenz vorhanden, aber was ist das schon? Warum macht kein Entwickler mehr aus seinen Sourcen? Wie wäre es mit einem Syntax-Checker, mit Analyseprogrammen oder mit einem syntaxgesteuerten Editor? Alles Dinge, die mit einem selbstgeschriebenen Compiler einfach zu realisieren wären.

## Trotzdem...

Bei allen offenen Wünschen, die vielleicht übersteigert aussehen mögen, sind ST-Besitzer sehr gut mit Modula-Compilern versorgt. Für fast alle Anwendungsschwerpunkte bietet der Markt genügend Auswahlmöglichkeiten.
C ist und bleibt eine feste Bank bei den Entwicklungspaketen, doch langsam aber sicher ist Modula-2 der große Konkurrent geworden, wenn es um Programmierung geht.

*RT*

*ENDE*

TEST

# DISKETTENFORMATE NACH MAß -

## DISKMASTER ST

**Sie haben ein hyper-super-duper-megamäßiges, überschlaues Primzahlenprogramm ("42 ist eine Primzahl") im unheimlich angesagten ST-BASIC geschrieben und wollen es jetzt geldbeutelfüllenderweise unter die Leute bringen? Sie suchen dazu noch einen möglichst fiesen Kopierschutz? Dann habe ich was für Sie: Diskmaster ST, eine bunte Mischung aus Trackmonitor und softer Formatiermaschine, stößt in einen bisher dünn besiedelten Markt vor: Formatierprogramme gibt es zwar mittlerweile mehr als genug, auch Diskmonitore sind keine Mangelware; allerdings beherrschen sie meistens nur ziemlich stinknormale Sektor-Operationen. Wer den Lesekopf seiner Floppy bei der Arbeit bespitzeln will, hat da schon weniger Auswahl.**




*Bild 1: So begrüßt Sie der Diskmaster*

Diskmaster ST tritt einem hilfreich zur Seite, wenn man/frau/hacker Diskettenformate und Kopierschutztricks entwerfen und auf die magnetbeschichteten Plastikschlaffis bannen will. Nebenher fördert das Programm den Durchblick durch grundlegende Funktionen eines Trackmonitors.

### Roots

Diskmaster ST ist ziemlich neu und andererseits auch wieder nicht. Stefan Wachter, der Autor, hat sich schon einmal - auf den guten alten 8-Bit-ATARIs - an einem ähnlichen Programm versucht. Die Erfahrung daraus (immerhin stecken im ST und in den Diskettenlaufwerken für die 8-Bit-ATARI-Serie fast identische Floppycontroller) konnte er jetzt gut verwerten, um den Diskmaster auf den ST zu portieren. Daß der gute Wachter von den 8-Bittern kommt, merkt man seinem Programm allerorten an - durchaus nicht immer nachteilig.

Den Intro-Bildschirm sehen Sie in Bild 1 (erinnert Sie das nicht auch an die 8-Bitter?). Was Sie nicht sehen können, ist der blitzschnelle Bildschirmaufbau: Diskmaster ST verwendet selbstprogrammierte Ausgaberoutinen. Beim Scrolling im integrierten Editor ist es wahrscheinlich sogar noch einen Hauch schneller als das legendäre TEMPUS. Allerdings zahlt man beim Diskmaster ST dafür auch einen Preis - die GEM-Einbindung ist dabei flöten gegangen. Stattdessen hat der Autor das Programm in selbstgestrickte Menügewänder eingekleidet, die sich nach einer Weile aber auch fast so heimelig anfühlen wie die Dropdowns (huch, es angelsächselt!) im DESKTOP. Diskmaster läuft übrigens nur im Monochrom-Modus, dafür aber auf allen Rechnern mit mindestens 512 KB Speicher und allen TOS-Versionen. Daß Diskmaster ST nicht nur beim Bildschirmaufbau unverschämt fix ist, hat seinen Grund: Die 65 KB des Programms sind in purem Assembler geschrieben - das ist allein schon eine starke Leistung. Ich schätze, der Quelltext ist mehr als ein halbes Megabyte groß - und das in Assembler. Ich ziehe meinen nicht vorhandenen Hut.

### Trackmonitor-Funktionen

Den Teil des Disketten-Maestros, der sich mit Sektor-Operationen beschäftigt, kann man getrost vernachlässigen. Zwar kann man Sektoren (auch solche mit Nicht-Standardgrößen) einlesen und

schreiben; wer aber erwartet, damit einen Standard-Sektormonitor ersetzen zu können, liegt falsch. Das will Diskmaster ja auch gar nicht sein. Immerhin kann man die Sektoren edieren (der Editor ist schnell, funktionsarm und griffig) und auch Teile ausdrucken.
Interessanter ist da schon, ganze Spuren einzulesen und anzuschauen (Bild 2). Auf diese Weise lernt man, wie eine Spur auf der Diskette aussieht - wenn auch die vielen Controllerfehler das Ergebnis (wie auch in Bild 2) stark verzerren. Das Verständnis wird etwas dadurch erleichtert, daß das Handbuch auf das Grundwissen zum Spuraufbau eingeht. Ein gutes Buch wird man sich trotzdem besorgen müssen, wenn man wirklich alles verstehen will, was im Reich des Lesekopfes vor sich geht (zur Not täte es ja auch ein Kurs wie die 'Floppyspielereien' in der ST-Computer).
Zwar könnte man jetzt natürlich Byte für Byte in der Spur ändern, um so ein Format aufzubauen - mühsam. Drum spart sich Diskmaster auch die Möglichkeit, eine Spur direkt zu schreiben; es gibt weniger steinige Wege nach Rom.

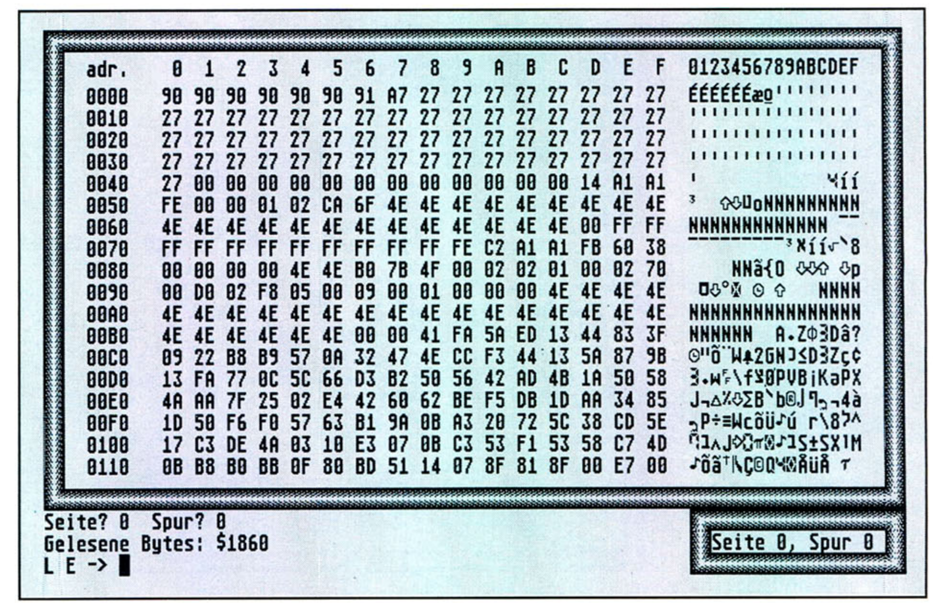



*Bild 2: Eine Diskettenspur unter der Diskmaster-Lupe*

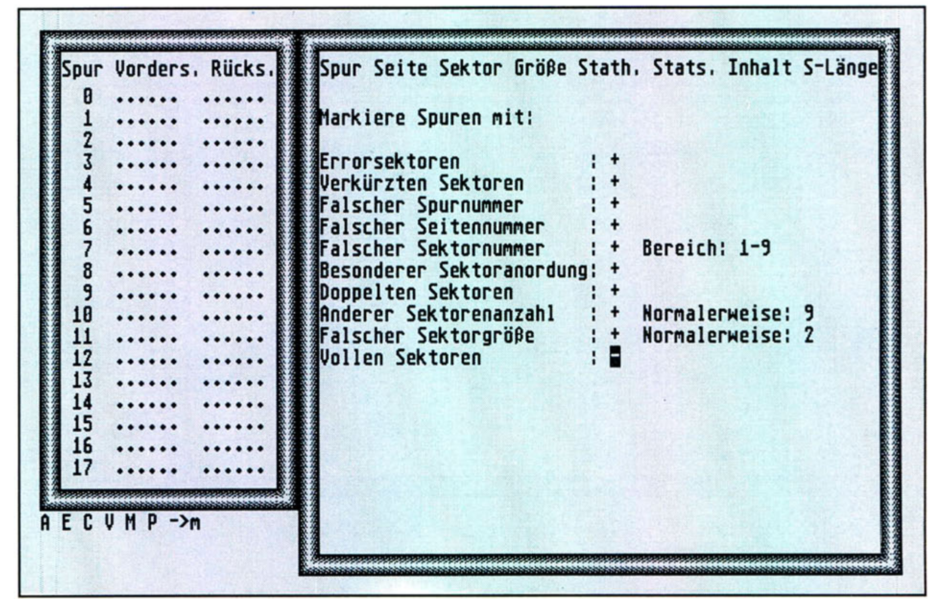


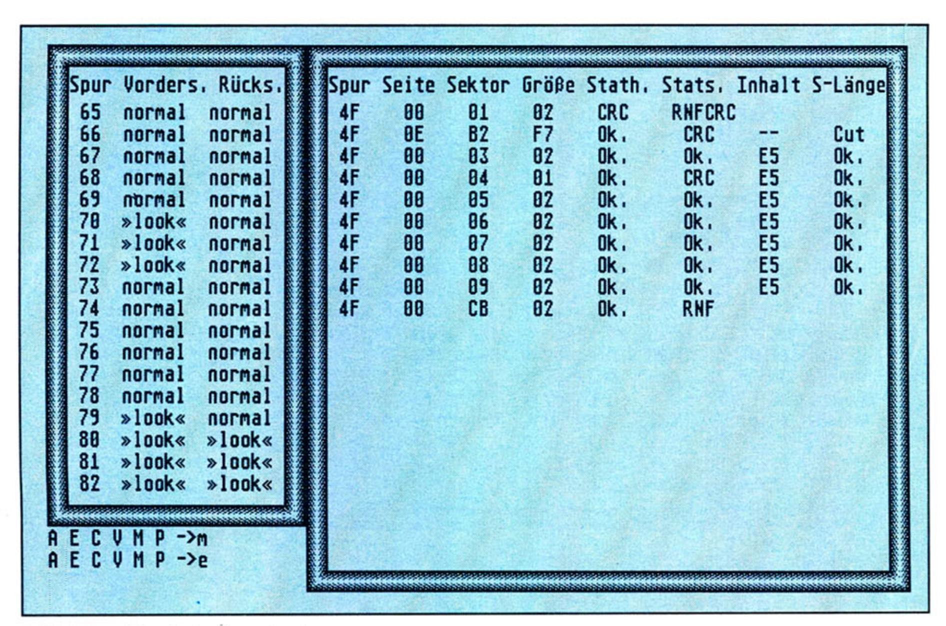

*Bild 3: Welche Eigenheiten soll Diskmaster bei der Diskettenanalyse anmahnen?*

## Drum prüfe, wer sich ewig schindet - die Analyse

Hat man eine tückische Diskette vor sich, kann man sie mit dem Diskmaster analysieren. Die Analyse geht nicht so flott vor sich wie fast alles andere in diesem Programm, aber wer sich schon einmal mit dem Floppycontroller WD1772 des ST beschäftigt hat, kann das gut verstehen: Um herauszufinden, was auf einer Spur steht, reicht es eben nicht, einfach die Spur einzulesen. Gesegnet seien die Jünger vom Stamme Western Digital...
Vor der Analyse souffliert man Diskmaster, auf welche Kriterien ganz besonders geachtet werden soll (Bild 3); danach kann man die Analyse starten. Das Ergebnis betrachtet man sich Spur für Spur auf dem Schirm (Bild 4) oder im Gesamtüberblick auf dem Drucker. Nicht immer ist sich Diskmaster bei der Analyse ganz sicher; es meldet sich dann mit der Bemerkung "velo", was wohl "verify and look" bedeuten soll.

## Formate selbstgezimmert

Der zentrale Gag des Diskmasters ist aber: In einer eigenen einfachen Programmiersprache bastelt man sich mit Hilfe eines integrierten Editors Beschreibungen von Diskettenformaten, die man übersetzen und ausführen (also auf die Diskette bringen) kann. Ein Beispiel sagt mehr als 1000 Worte (Listing 1):
Im Listing 1 wird ein Format definiert, das 30 Sektoren mit jeweils 128 Bytes auf eine Diskettenspur bringt. Wie Sie sehen, bietet die Format-Sprache Konstrukte wie eine einfache Schleife (LOOP x - ENDLOOP), außerdem Funktionen zur Verwaltung von Tabellen (für repetitive Aufgaben). Man kann beliebige Zahlen oder auch Strings in den Format-Puffer übernehmen (ZAHL, ASCII, FILL, TABFILL), auch Zufallszahlen (ZUFALL) oder aktuelle (vordefinierte) Laufvariablen, die Diskmaster selbst verwaltet (SPUR, SEITE). Wer sich vor Programmiersprachen gar zu gerne drückt, kann Formate auch in Form einer Tabelle definieren (Bild 5).
Leider bietet der Diskmaster in dieser Maske nur Platz für 28 Sektoren. Wer mehr braucht, muß sich dann doch zum Programmieren bequemen.
Die Format-Definition ist elegant und einfach. So kann man sich ohne großes Aufhebens seinen eigenen Kopierschutz ausdenken und austesten. Ein paar (grundlegende) Anregungen dafür finden sich im Handbuch. Darüberhinaus werden Assemblerroutinen mitgeliefert, die ausreichen, um die im Handbuch beschriebenen Kopierschutztricks abzufragen. (Was hülfe auch ein Schutz ohne Abfrage?). Die ganz kniefiesligen, relativ neuen Tricks beherrscht der Diskma-



| Spur | Vorders. | Rücks. |
|---|---|---|
| 65 | normal | normal |
| 66 | normal | normal |
| 67 | normal | normal |
| 68 | normal | normal |
| 69 | normal | normal |
| 70 | »look« | normal |
| 71 | »look« | normal |
| 72 | »look« | normal |
| 73 | normal | normal |
| 74 | normal | normal |
| 75 | normal | normal |
| 76 | normal | normal |
| 77 | normal | normal |
| 78 | normal | normal |
| 79 | »look« | normal |
| 80 | »look« | »look« |
| 81 | »look« | »look« |
| 82 | »look« | »look« |

| Spur | Seite | Sektor | Größe | Stath. | Stats. | Inhalt | S-Länge |
|---|---|---|---|---|---|---|---|
| 4F | 00 | 01 | 02 | CRC | RNFCRC | | |
| 4F | 0E | B2 | F7 | Ok. | CRC | -- | Cut |
| 4F | 00 | 03 | 02 | Ok. | Ok. | E5 | Ok. |
| 4F | 00 | 04 | 01 | Ok. | CRC | E5 | Ok. |
| 4F | 00 | 05 | 02 | Ok. | Ok. | E5 | Ok. |
| 4F | 00 | 06 | 02 | Ok. | Ok. | E5 | Ok. |
| 4F | 00 | 07 | 02 | Ok. | Ok. | E5 | Ok. |
| 4F | 00 | 08 | 02 | Ok. | Ok. | E5 | Ok. |
| 4F | 00 | 09 | 02 | Ok. | Ok. | E5 | Ok. |
| 4F | 00 | CB | 02 | Ok. | RNF | | |

A E C V M P ->m
A E C V M P ->e

*Bild 4: Das Ergebnis einer Analyse*

| Sektor | Error | Länge | Inhalt | Sektor | Error | Länge | Inhalt |
|---|---|---|---|---|---|---|---|
| 01 | 0 | 080 | CB | 10 | 0 | 080 | CB |
| 02 | 0 | 080 | CB | 11 | 0 | 080 | CB |
| 03 | 0 | 080 | CB | 12 | 0 | 080 | CB |
| 04 | 0 | 080 | CB | 13 | 0 | 080 | CB |
| 05 | 0 | 080 | CB | 14 | 0 | 080 | CB |
| 06 | 0 | 080 | CB | 15 | 0 | 080 | CB |
| 07 | 0 | 080 | CB | 16 | 0 | 080 | CB |
| 08 | 0 | 080 | CB | 17 | 0 | 080 | CB |
| 09 | 0 | 080 | CB | 18 | 0 | 080 | CB |
| 0A | 0 | 080 | CB | 19 | 0 | 080 | CB |
| 0B | 0 | 080 | CB | 1A | 0 | 080 | CB |
| 0C | 0 | 080 | CB | 1B | 0 | 080 | CB |
| 0D | 0 | 080 | CB | 1C | 0 | 080 | CB |
| 0E | 0 | 080 | CB | 1D | 0 | 080 | CB |
| 0F | 0 | 080 | CB | 1E | 0 | 080 | CB |

Gapbytes
4E 00
|| ||
1: 20 10
2: 10 00
3: 22 12

1 Nach Index
2 Vor Adress
3 Vor Daten

Format F

E N T F A K -> e

*Bild 5: Definition des 30-Sektor-Formats aus Listing 1 mit anderen Mitteln*

ster ST ohnehin nicht (nein, ich werde mich jetzt nicht darüber auslassen). Aber es reicht, um einem durchschnittlichen Cracker eine Weile die Zeit zu vertreiben. Die Abfrageroutinen kann man leicht in eigene Programme einbinden; Beispiele in Megamax C, ST Pascal, GST-Assembler, GFA-BASIC und Omikron-Basic finden sich auf der Programmdiskette.

## Von der Spur zur Diskette

Nun hat man also einige Spurformate mit mehr oder weniger trickreichem Unsinn darin entworfen; damit möchte man eine Diskette vollkleistern. Auch dafür bietet Diskmaster eine Lösung (Bild 6): Die einzelnen Spurformate können beliebig über eine Diskette verteilt werden.

## Eine Diskette geht in Serie

Das ist aber immer noch nicht alles! Auf die Funktionen des gesamten Programms ist eine weitere Programmiersprache mit 27 Befehlen aufgesetzt, von der aus man fast alle Teile des Programms steuern kann. Von einer Quelldiskette Sektoren lesen, sie in einem Puffer zwischenspeichern, sodann eine Zieldiskette formatieren (eventuell mit allerlei Kopierschutztricks) und die Informationen der Quelldiskette darauf abspeichern - kein Problem. Sehr praktisch: Die Sprache ist mächtig genug, um auch zufällige oder vorbestimmte, aber eindeutige Seriennummern auf die Disketten zu schreiben. Es ist also endlich auch für Otto Normalhacker kein Problem mehr, eine kleine Serienproduktion von Programmdisketten anzufertigen. Listing 2 ist ein Beispiel für einen solchen Produktionstext.

Die Produktionssprache versteht leider wie auch die Format-Definitionssprache keine arithmetischen Ausdrücke und kennt weder Variablen noch Schleifen, nur einen Sprungbefehl, der als Ziel eine Marke im Programmtext hat (man kann nur eine Marke setzen). Obwohl das recht restriktiv klingt, reicht es meistens aus, um auch vertrackte Programmdisketten zu produzieren. So kann man etwa alle vordefinierten Spur- und Diskettenformate "ausführen" (also formatieren), bestimmte Kopierschutzmethoden automatisch anwenden (doppelte Sektoren, gezielt zerstörte Sektoren, 18-Sektor-Format), einen Bootsektor ausführbar machen und mit Puffern jonglieren, die man für allerlei Transferzwecke gebrauchen kann.

Für die Produktion von Disketten empfiehlt sich ein zweites Laufwerk und mindestens 1 Megabyte Speicher; es läuft alles auch ohne diese Zutaten, doch läuft man dann Gefahr, sich einen Diskettenwechsel-Arm zu holen.

## Kleinkram

Unter diese Kategorie fallen viele Details: Drehzahlmessung für Laufwerk A und B, ein kleines mitgeliefertes Extra-Programm, das die berüchtigten CRC-Prüfsummen berechnet, Hilfstexte, die man sich in den meisten Situationen per HELP-Taste aufrufen kann, die Unterscheidung zwischen logischem Laufwerk (dort werden allerlei Puffer und die Programmtexte abgelegt) und physikalischem Laufwerk (dort wird tatsächlich Arbeit verrichtet)...

## Fehl und Tadel

Man muß sich schon daran gewöhnen, wie Diskmaster ST seine Spurformate, Diskettenformate und Produktionstexte auf der Diskette verwaltet; vor allem hat man im Programm selbst keine richtige Verzeichnisfunktion, bloß eine selektive, die nur die vom Diskmaster erzeugten Dateien anzeigt. Das wiegt umso schwerer, als man wegen des Verzichts auf GEM-Einbindung keine Accessories aufrufen kann, um sich zu behelfen. Wenn man dann aus dem Diskmaster heraus schnell nachsehen will, ob die eingelegte Diskette die Systemdiskette oder eine Leerdiskette ist (wer beschriftet seine schlaffen Scheiben denn heutzutage noch?), bleibt einem nur der Umweg

über die eher kärglichen Sektor-Lese-Funktionen des Diskmaster.

Der GEM-Verzicht verlangt einem auch ein bißchen Flexibilität bei der Bedienung ab; man merkt aber Diskmaster positiv an, daß sich der Autor bei der Bedienung einiges gedacht hat.

Beim Laden und Speichern von Puffern treten ab und an etwas eigenwillige Fehler auf, die laut Handbuch verschwinden, wenn man zwischendurch die Diskette im betreffenden Laufwerk wechselt (ein bißchen knifflig bei einer RAM-Disk, würde ich sagen). Nicht lebensgefährlich, zumal die Fehlermeldungen im (knappen) Klartext ausgegeben werden, aber äußerst unschön.

Die Programmiersprachen für Spurformate und komplette Produktionen sind noch etwas eingeschränkt; vor allem fehlt die Behandlung von einfachen arithmetischen Ausdrücken, die ich mir ab und an beim Programmieren gewünscht hätte. Besonders bei der Produktionssprache vermisse ich bedingte Verzweigungen. Sicher, man kommt in den allermeisten Fällen auch ohne diesen "Luxus" aus; bisweilen muß man dann aber doch eine Diskette nach der Produktion nachbehandeln, womit sich der Nutzen des Programms erheblich relativieren kann.

Das Handbuch ist ausreichend, dabei knapp und meistens präzise. Nicht jeder aber hat wie ich schon eine ganze Weile mit Disketten herumgespielt; dem Anfänger in Sachen Floppy wäre mit einem kleinen Beispiel bei jedem Befehl der aufgesetzten Programmiersprachen sehr gedient. Einige Ungenauigkeiten im Handbuch sind mir auch aufgefallen. So behauptet der Autor von bestimmten vorgestellten Kopierschutzmethoden, daß sie mit diesem oder jenen Kopierprogramm nicht zu knacken seien - was ich in mindestens zwei Fällen widerlegen konnte. 11-Sektor-Formate sind auch nicht - wie angegeben - generell um die Hälfte langsamer (siehe 'Floppyspielereien I', ST-Computer 6/87) und auch nicht unbedingt gefährlicher - es kommt eben aufs Laufwerk an.

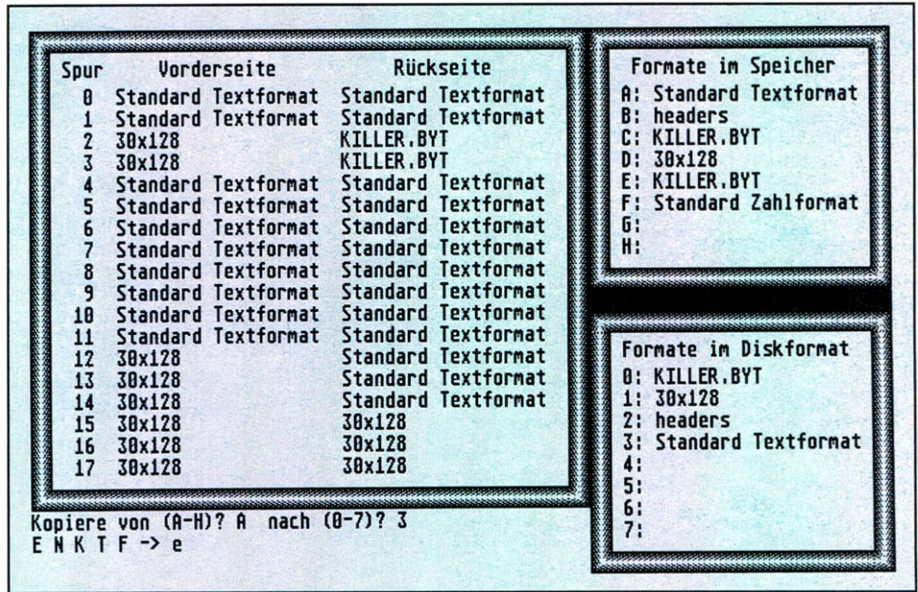


*Bild 6: Eine Diskette wird verplant*

## Alles in allem

Diskmaster ST ist eine Möglichkeit für den Einsteiger, sich mit Disketten zu befassen. Dabei helfen die Trackmonitor-Funktionen und die wirklich gute Idee, ein Format durch eine Sprache zu beschreiben. Die wirkliche Zielgruppe aber, das sind wohl diejenigen Zeitgenossen, die Programme selbst vermarkten wollen. Es soll aber keiner denken, ohne zusätzliche Dokumentation und Anregung könnte man nun am Fließband wirklich hinterlistige Schutzmethoden austüfteln - das Handbuch reicht dazu nicht aus (wäre vielleicht auch ein bißchen viel verlangt); dazu braucht man immer noch einen Schuß Phantasie und gehörige Sachkenntnis.

Ja, und dann habe ich da noch so ein Problem: Ich kann Kopierschutz nicht ausstehen. Und Diskmaster ST trägt ja die Werkzeuge dazu in jedes Hackerzimmer. Ich kann mich nur damit trösten, daß so auch die "Gegenseite" informiert wird, so daß der eine oder andere Schutz wohl bald wertlos sein wird. Wohlgemerkt: Ich rede hier nicht der Raubkopie das Wort. Ich glaube eben nur, daß letzten Endes ein gutes Handbuch, ein reibungsloser Update-Service und eine Seriennummer den besten Kopierschutz dar-

```
 1:  * Spezialformat mit 30 Sektoren mit jeweils
 2:  * 128 Bytes
 3:
 4:  deftab 0    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
 5:              19 20 21 22 23 24 25 26 27 28 29 30
 6:  * Tabelle mit Sektornummern
 7:
 8:  fill 40 $4e                   * Spurvorspann (40 * $4E)
 9:
10:  loop 30                       * 30mal durch die Schleife
11:
12:     fill 10 $4e                * Sektorvorspann (10 * $4E)
13:
14:     zahl $f5 $f5 $f5 $fe       * Syncmarke für Adreßfeld
15:     spur
16:     seite                      * Spur und Seite einfügen
17:     tab 0                      * nächste Sektornummer aus
18:                                * Tabelle 0 holen
19:     zahl 0                     * Sektorlänge 0 (128 Bytes)
20:     zahl $f7                   * Prüfsumme schreiben
21:
22:     fill 22 $4e                * Post Address Mark Gap
23:     fill 12 0                  * (22*$4e, 12*0)
24:     zahl $f5 $f5 $f5 $fb       * Syncmarke Datenfeld
25:     fill 128 $e5               * Sektorinhalt (128*$E5)
26:     zahl $f7                   * Prüfsumme schreiben
27:
28:     fill 10 $4e                * Sektornachspann (10*$4E)
29:
30:  endloop                       * nächster Sektor
31:
32:  fill 200 $cb              * kleiner Spurnachspann (200*$CB)
33:
34:  =====================================================
```

*Listing 1: Ein Format im Eigenbau*

```
 1:  * Beispiel für einen einfachen Produktionstext
 2:
 3:  lade_s_format 'killer.byt'       * vordefinierte
                                        Spurformate
 4:  lade_s_format 'short30'          * vom logischen
 5:  lade_s_format 'headers'          * Laufwerk
 6:  lade_s_format 'standard'         * lesen
 7:
 8:  taufe_puffer 'secbuf'            * Puffer anlegen
 9:  print '*** Start-Seriennummer eingeben:'
10:  input_s_nummer                   * Start-Seriennummer
11:                                   * einlesen
12:
13:  marke                            * Sprungmarke setzen
14:
15:  setze_laufwerk A                 * Ab jetzt alles auf
16:                                   * Laufwerk A
17:
18:  format 0-1 0-70 'standard textformat'
19:  format_18 0-1 71
20:  format 0-1 72 '30x128'
21:  format 0-1 73 'headers'
22:  format 0-1 74 'killer.byt'
23:  format 0-1 75-79 'standard textformat'
24:  * Diskette ist jetzt formatiert, der Schutz ist
25:  * auf den Spuren 71-74
26:
27:  setze_laufwerk B                 * von Laufwerk B
28:  lese_daten 0-1 0-70 1-9          * Programmdaten lesen
29:
30:  setze_laufwerk A                 * auf Laufwerk A
31:  schreibe_daten 0-1 0-70          * PRG-Daten schreiben
32:
33:  lese_puffer 0 0 1 'secbuf'       * Bootsektor einlesen
34:  fülle_puffer 8-10 257 'secbuf'   * zufällige TOS-
35:  poke_s_nummer 500 'secbuf'       * Seriennr. und
36:                                   * Produktions-Serien-
37:                                   * nummer einfügen
38:  erhöhe_s_nummer                  * Seriennummer + 1
39:
40:  bootbar 1 'secbuf'               * Bootsektor ausführbar
41:  schreibe_puffer 0 0 1 'secbuf'   * machen und schreiben
42:
43:  lese_puffer 0 79 2 'secbuf'      * Sektor 2 auf Spur 79
44:  kill_sektor 0 79 2 $20 'secbuf'  * (Seite 0) zerstören
45:
46:  print '*** Ich bin völlig am Ende...'
47:  taste
48:
49:  springe                          * zur Marke springen
50:
51:  =====================================================
```

*Listing 2: Diskettenproduktion im Akkord*

stellen. Alle anderen Tricks behindern nur die ehrlichen Anwender - wie auch beim Diskmaster ST: Wenn man das Programm von RAM-Disk oder Platte startet und die Original-Diskette nicht in Laufwerk A liegt, kommentiert das der Diskmaster trocken mit "Ihr Rechner hat nicht genug Speicher" - und das bei einem MegaST2.

Trotzdem, das Programm an sich ist so durchdacht, sauber aufgebaut, schnell und praktisch, daß ich es auf jeden Fall weiterempfehlen muß. Zumindest denjenigen, die kleine bis mittlere Mengen von Programmdisketten produzieren müssen und sich einen Haufen Arbeit und Zeit sparen wollen.

*CB*

*Diskmaster ST kostet 79 DM und ist zu haben bei:*

*Stefan Wachter*
*Haslacher Weg 45*
*7900 Ulm*

**ENDE**

## A-NET
## DAS NETZWERK FÜR ATARI

Verbinden Sie Ihre Atari Computer störungsfrei mittels *modernster Lichtwellenleitertechnik*. Greifen Sie auf gemeinsam genutzte Daten und Ressourcen zu!
Und das alles mit der gewohnten Einfachheit der DESKTOP Operationen!

**A-NET Grundeinheit** 1650.-
ein Master- u. zwei Slave-interfaces, 15 m LWL, Software

**FLASH ACCESS** a.A.
Multiuser Datenbank für Softwareentwickler

**PRIVATLIQUIDATION ST** 525.-
Abrechnung nach GOÄ/GOZ, Patientenverwaltung, Mahnung, Kostenstellen.
(für Tierärzte in Vorbereitung)

**LOHN & GEHALT ST** 725.-
GEM unterstützte Bruttolohn- und Nettolohnabrechnung, Auswertung, Formulare, Überweisungen.

DM COMPUTER

**DM COMPUTER GMBH**
Durlacherstraße 39
7530 Pforzheim
Tel.:07231-13939
Tlx.:783248

**Professionelle Finanzbuchhaltungsprogramme**

| | |
|---|---|
| Mini-Lern-Fibu 1.0 | 98,– DM |
| ST-Fibu 1,5 | 298,– DM |
| Mandantenfähig | 398,– DM |
| ST-Fibu 2.0 | 498,– DM |
| Mandantenfähig | 649,– DM |

**GEORG STARCK**
**Herzbergstr. 8 · D-6369 Niederdorfelden**
**Tel. 06101-3007 Hotline bis 22.00 Uhr**

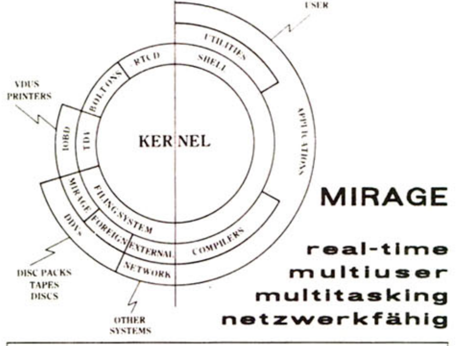
**HSS**

Hardware-Software Systemlösungen
Harmel-Scollar & Schmithals GbR
**Darmstädter Str. 20 · 5000 Köln 1 · ☎ 0221/316207**

Der Partner für Handwerker, Kleinbetriebe und Handel. Preiswerte maßgeschneiderte EDV-Systeme durch unsere unabhängige Beratung.

| | | |
|---|---|---|
| **Cash Soft** | **T.I.M. (Buchführung** | **229,-** |
| **Angebot!** | **Cashflow (Kassenbuch)** | **229.-** |
| **Bavaria Soft** | BSS-Plus Handwerkspaket | a.A. |
| | BS-Handel | 479,- |
| | BS-Fibu | 579,- |
| **Prodata** | fibuMAN e | 379,- |
| | fibuMAN f | 749,- |
| **Star Division** | Star Writer ST | 189,- |
| **Atari** | Atari SH 205 | |
| | + Backup Utility | 1098,- |
| **Vortex** | HDPLUS 30 | 1298,- |
| | HDPLUS 60 | 2198,- |

Weitere Hardware von Atari, Star, Nec, Tandon und sämtliche Atari Software auf Anfrage.
Versandbedingungen: Preise inkl. Versandkosten (Scheck), bei Nachnahme zuzügl. DM 5,-, alle Preise für Atari ST-Version.




## MIRAGE

**real-time**
**multiuser**
**multitasking**
**netzwerkfähig**

**Die Alternative zu UNIX.** Seit Jahren hat **MIRAGE** sich auf dem Markt der Workstations fest etabliert. Jetzt gibt es die ST-Version. Extrem schnell. Extrem vielseitig. Einige Features:

- Lesen & Schreiben von TOS-Disketten
- TOS-und **MIRAGE**-Partitions auf **einer** Festplatte
- TOS-Calls von **MIRAGE** aus möglich (**kein** GEM)
- Line-A Grafik zugänglich
- virtuelle Bildschirme (umschaltbar per Tastatur)
- bis zu 128 dynamische Tasks pro Job
- Kommandosprache mit Variablen, Nesting etc.
- geräteunabhängiges Treiberkonzept
- schneller, komfortabler Editor
- Supervolumes: Eine Datei über mehrere Drives
- echtes Netzwerk ohne zusätzliche Hardware
- Sprachen: Basic, Pascal, Lisp, Fortran, C, APL

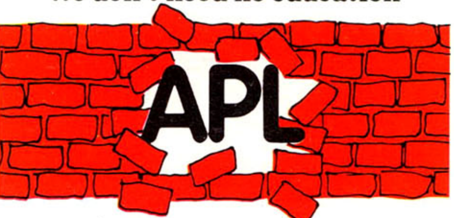
Der **Assembler**:
- Macros (mit lokalen Labels und Konstanten!)
- bedingte Assemblierung
- Unterstützung von 68020/68030/68881/68882
- Fließkomma-Konstanten
- Linker, Debugger, Profiler, Library-Funktionen

**MIRAGE:** Die ideale Lösung für Software-Entwickler, die eine preiswerte Workstation-Umgebung brauchen.
**MIRAGE:** Für Unis, die z.B. ihre Fortran-Anwendungen aus dem Rechenzentrum lösen wollen (Swifte-Fortran -77 ist als **Error-Free** zertifiziert, die **NAG**-Bibliotheken sind erhältlich - auch auf Disketten!).
**MIRAGE:** Für Meßdatenerfassung & Steuerzwecke.
**MIRAGE:** Für die Portierung von PC-Software auf den ST (z.B. mit Lattice C); für Netzwerke; ...

**MIRAGE-ST** (ROM-Pack, Disks, Manuals) **348.-**
**MicroGrid** (Netztreiber für RS232 u. MIDI) **98.-**
- Bitte fordern Sie Infos und Preislisten an -

**gdat** Stapelbrede 39 4800 Bielefeld 1 **0521/875 888**

We don't need no education




We don't need no thought control

**APL/68000** - Das ganz besonders schnelle APL für 68000er Rechner. **Von MicroAPL.** Die Sprache der Profis für ATARI, MAC und AMIGA - oder darf es eine Workstation sein? Bei **gdat** zum supergünstigen Preis: Incl. Support, Update-Service und Toolsdiskette (nur ST). **Wir liefern immer die neueste Version.** Literatur- und Referenzhinweise sowie ausführliche Infos gratis.

**APL/68000-ST+** **298.-**
Die PLUS-Version wird exklusiv von **gdat** vertrieben und enthält wertvolle Erweiterungen und Tools, darunter sogar eine kleine Datenbank!

**APL/68000 Language Manual** **49.-**
Der Kaufpreis wird bei nachfolgendem Kauf des APL voll angerechnet.

**APLPRINT** **198.-**
druckt Text und Grafik unter APL.
**LineAl** **164.-**
Schnelle Line-A Grafik.
**APL-ASS** **164.-**
Für Assembler-Routinen in APL.
**APL-EDIT** **248.-**
Variablen-Editor, Icons, Fonts ...
**GD_GRAPH** **48.-**
VDI-Grafiktools.

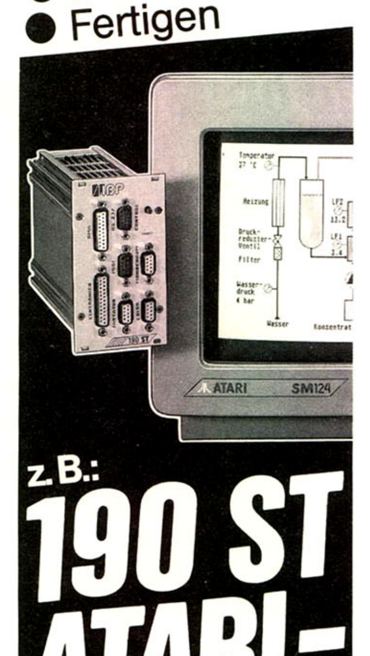
APL ist eine Hochsprache der vierten Generation, die seit Jahren ihre Effizienz erwiesen hat. Und: APL ist eine Sprache zum Anfassen - Leicht zu lernen, intuitiv, fehlertolerant. Mit APL werden Probleme gelöst, nicht geschaffen. Zudem können wir uns als zuverlässigen Partner empfehlen: **gdat** arbeitet seit 7 Jahren mit APL/68000. Wir wissen, wovon wir reden. Fordern Sie das Info an.

**gdat** Stapelbrede 39 4800 Bielefeld 1 **0521/875 888**




Messen, Steuern, Regeln

- Beraten
- Entwickeln
- Fertigen




z.B.:
# 190 ST ATARI-
**kompatible Industrie-Lösung**

Auf Basis des Atari/ST-Konzeptes und unter Verwendung von Atari-Bauelementen hat IBP eine Industrieausführung entwickelt.

Besondere Merkmale:
- Buserweiterungen (EUROBUS, ECB, VME) integriert, daher vielfältig ausbaubar.
- sehr kompakte, mechanisch-stabile Bauweise
- 19"-Rack (3HE/14TE)
- volle Atari/ST-Hard- und Software Kompatibilität

IBP bietet eine Fülle von Karten AD/Input/Output für verschiedenste Aufgaben und komplette Problemlösungen.

**IBP**

IBP Gerätebau GmbH
Lilienthalstraße 13
3000 Hannover 1
Telefon 0511/630963/64




## Ein preiswertes CAD-Programm mit guten Fähigkeiten

**Das Einsatzgebiet von CAD-Programmen ist sehr groß. Es reicht von technischen Kontruktionszeichnungen über elektrische Schaltungen und Platinenlayouts bis zum Einsatz im Architekturbereich. Aber auch für die Erstellung von Programmablaufplänen oder Struktogrammen sind sie für Programmentwickler eine große Hilfe.**

CAMUS DRAFT ist ein solches CAD-Programm, das im ersten Moment vor allem durch seinen sehr niedrigen Preis auffällt.
Kommerzielle CAD-Systeme bewegen sich in Preisregionen, die für private Anwender völlig unerschwinglich sind und auch für den ATARI ST muß man, je nach Qualität des Programms, noch einige hundert Mark ausgeben, um ein Programm mit CAD-Fähigkeiten zu erwerben. CAMPUS DRAFT ist mit DM 149.- ein ausnahmslos günstiges Programm. Deshalb stellt sich die Frage, wo es einzuordnen ist.

## Bedienung und Leistung

Schon gleich nach dem Starten fällt die Ähnlichkeit zur 'großen Version' CAMPUS CAD auf. Dies betrifft vor allem die Bedienung über die Menüleiste und die am linken Rand angeordneten Symbole. Die Symbole stellen dabei eine der Stärken des Programms dar, denn sie ersparen die Suche nach den Funktionen in den Menüs und tragen sehr zum Bedienungskomfort bei.
Der Symbolbereich ist in fünf Gruppen unterteilt, in denen sich artverwandte Funktionen befinden. Im ersten Teil findet man die globalen Funktionen, die sich auf die gesamte Zeichnung beziehen.

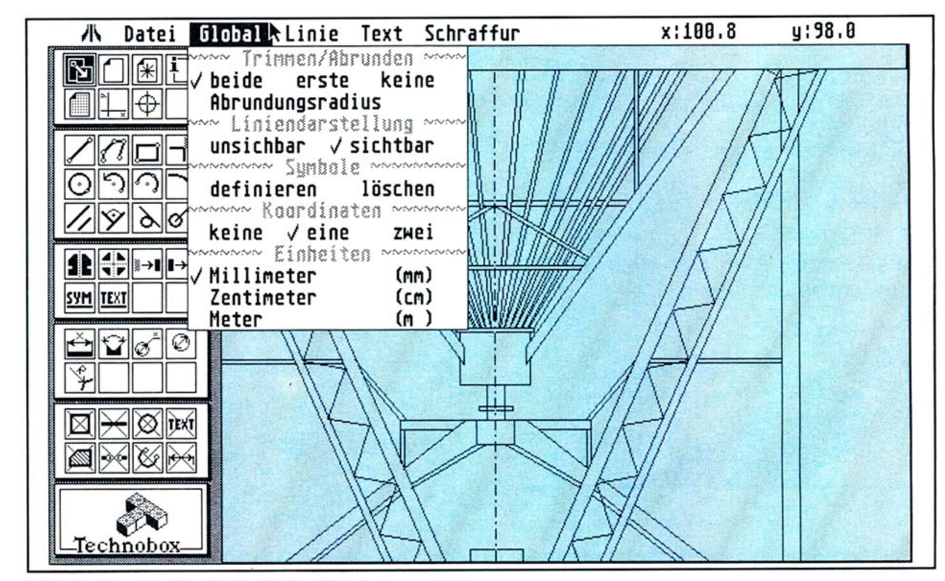



*Bild 1: Campus Draft, der kleine Bruder*

Dazu gehört das stufenlose Zoomen, das jedoch nur in einer Richtung zu betreiben ist. Will man die Vergrößerung wieder etwas zurücknehmen oder einen anderen Ausschnitt wählen, muß man immer wieder über die Funktion 'volles Format' das Gesamtbild aufbauen lassen. Diese Einschränkung ist beim Arbeiten sehr störend und auf die Dauer unbefriedigend. Zwar baut CAMPUS DRAFT recht schnell ein Bild auf, aber trotzdem vergeht einige Zeit, bis man sich an die richtige Stelle gezoomt hat.
Eine weitere wichtige Funktion dieser Gruppe ist das Setzen eines Rasters. Dies bedeutet bei einer Konstruktion eine wertvolle Hilfe, denn nun bewegt sich das Fadenkreuz nur noch auf den Rasterpunkten und erleichtert somit die Positionierung.
Allerdings sollte man sich die geeigneten Werte des Rasters gleich zu Beginn der Zeichnung überlegen und sie einstellen, um spätere Komplikationen zu vermeiden.
Wenn man nicht mit dem Raster arbeitet, ist es meistens sinnvoll, die Fangradius-Funktion einzuschalten, die innerhalb eines einstellbaren Bereichs Linienenden aufeinanderzuführt.
Die zweite Symbolgruppe beinhaltet die Grundfunktionen wie Linie, Polygonzug, Rechteck, Vollkreis, Kreisbogen, Kreis durch drei Punkte, Parallele, Lot, Tangente, Normale und Abrunden zweier Linien.
Die nächste Gruppe bilden die Funktionen, die sich jeweils auf einen definierten Bereich auswirken. Die Funktion Spiegeln ist hierbei die einfachste; beim Kopieren und Verschieben kann zusätzlich noch ein Vergrößerungsfaktor bestimmt werden. Am leistungsfähigsten ist das Rotieren, denn hier kann neben dem Rotationswinkel noch die Anzahl der Kopien vorgegeben werden, so daß hiermit sehr einfach komplexe Operationen ausgeführt werden können.
Ebenfalls in dieser Gruppe ist der Text-Befehl, mit dem Texte in die Zeichnung eingefügt werden können. In einem speziellen Menü lassen sich dafür die Texthöhe, der Ausrichtungswinkel (0'-360') und einige Schriftattribute setzt.

Ein spezieller Bereich im Symbolfeld ist der Bemaßung vorbehalten. Strecken-, Winkel-, Kreis-, Durchmesser- und Radienbemaßung sind die vorgesehenen Punkte. Die Ausführung dieser Funktionen ist recht einfach, denn man muß nur zwei Bezugspunkte mit der Maus anwählen und dann die Textplazierung wählen, woraufhin Bemaßungslinien, Pfeile und Werte automatisch eingetragen werden. Vermissen wird man vielleicht die Möglichkeit zur Kettenbemaßung, obwohl sie mit etwas Handarbeit auch zu bewältigen ist. Daß bei kurzem Bemaßungsabstand die Bemaßungspfeile trotzdem innerhalb dieser Linien angesetzt werden führt zu unschönen Überschneidungen. Auch hier hilft nur Handarbeit.

Die letzte Symbolgruppe enthält ausschließlich Funktionen zum Löschen. Eine schraffierte Fläche oder eine komplette Bemaßung kann somit durch einen Mausklick aus der Zeichnung entfernt werden.

CAMPUS DRAFT verarbeitet Zeichnungen im Format A5-A2 mit bis zu 9 Ebenen haben, die wahlweise ein- bzw. ausgeblendet werden können.

Damit können z.B. Layouts einfach getrennt werden, zudem beschleugt sich der Bildaufbau, wenn nur einige der benutzten Ebenen eingeblendet werden.

Alle für technische Zeichnungen gängigen Linienart (6) sind in den Strichstärken 0.25, 0.35, 0.5 und 0.7 vorhanden. Allerdings ist es während des Zeichnens nicht sinnvoll die wahre Strichstärke darzustellen, denn dann nimmt der Bildaufbau sehr viel Zeit in Anspruch. Das Zeichnen mit einer einzigen Linienstärke verlangt vom Benutzer allerdings einiges an Disziplin, zumal die aktuelle Stärke nicht eingeblendet wird.

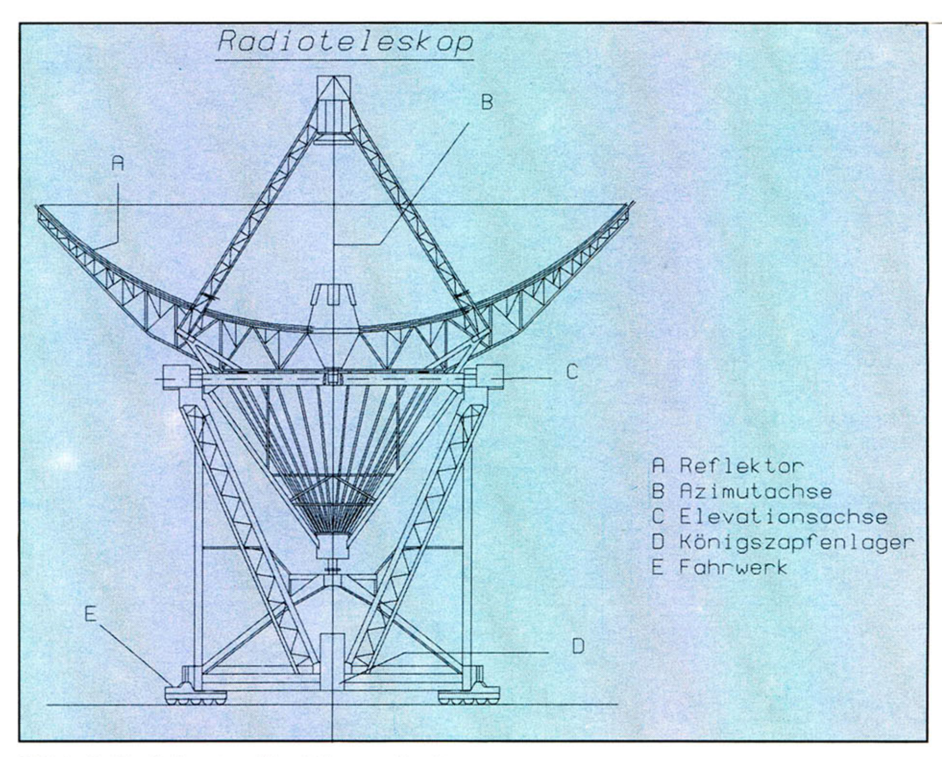



*Bild 2: Radioteleskop, erstellt mit Campus Draft*

## Schraffieren

Beim Zeichnen am Zeichenbrett ist besonders das Schraffieren größerer Flächen eine monotone und zeitraubende Angelegenheit nicht so mit einem CAD-Programm. Nachdem man den Schraffurwinkel und den Linienabstand eingestellt hat, genügt ein Klicken mit der Maus, um selbst große Flächen schnell mit einem Muster zu versehen.

CAMPUS DRAFT verwendet zur Definition der Konturen eine gruppenorientierte Methode, um Probleme, die sonst bei der 'Konturverfolgung' auftreten können, zu umgehen. Diese Methode bringt den Vorteil, daß auch nicht vollständig geschlossene Flächen bearbeitet werden könnnen. Die bereits ausgewählten Konturelemente werden dabei hell dargestellt, so daß man jederzeit einen Überblick hat, welche Fläche für die Schraffur ausgewählt worden sind. Dabei ist es auch möglich, Schraffurflächen ineinander zu verschachteln und somit bestimmte Teile von der Schraffierung auszusparen.

## Symbolverwaltung

Wenn man Zeichnungen erstellt, kommt es häufig vor, daß man die gleichen Elemente mehrmals benötigt. Deshalb ist es praktisch und zeitsparend, wenn man diese Elemente festlegen und speichern kann, so daß sie auf Tastendruck zur Verfügung stehen. Um ein Symbol zu erstellen, zeichnet man es und rahmt diesen Bereich ein. Dieses Element wird dann unter einem beliebigen Namen in einer Symboldatei abgespeichert und steht dann in einer Auswahlbox zur Verfügung.

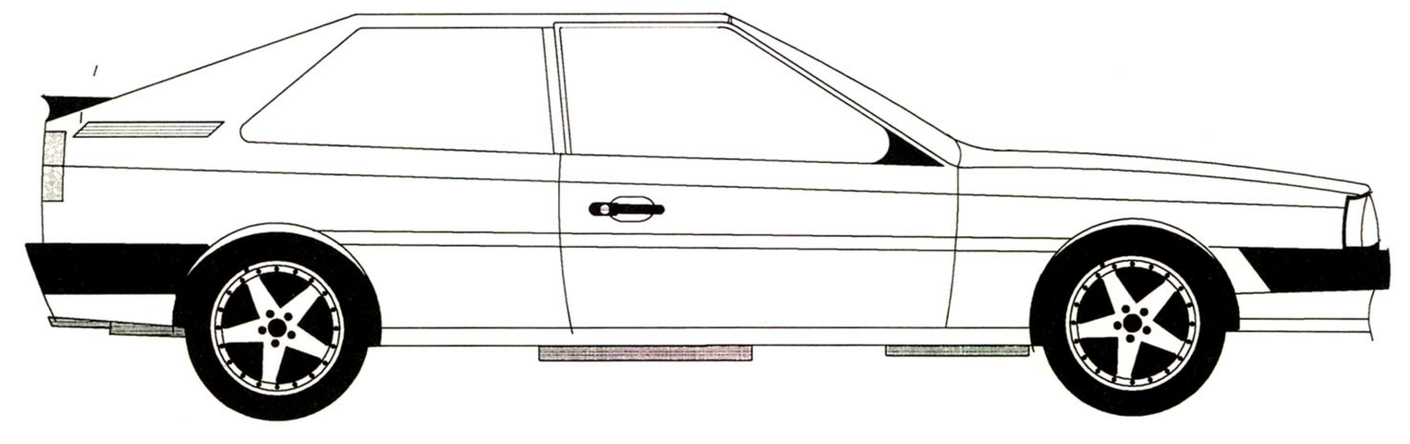


*Bild 3: Laserausdruck einer Campus Draft-Zeichnung*

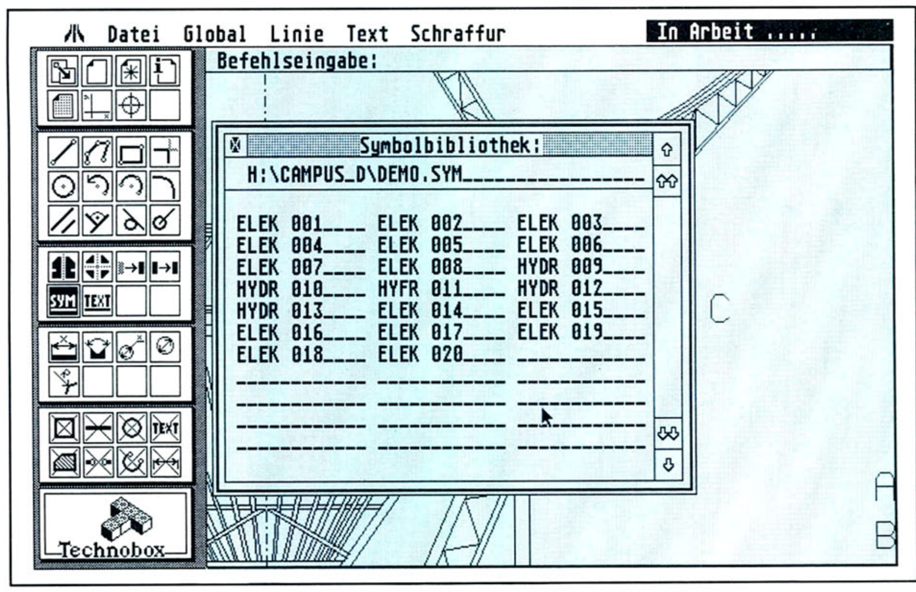



*Bild 4: Eine Nutzung von Symbolbibliotheken ist auch möglich*

## Drucken - Plotten - Lasern

Um eine Zeichnung ausgeben zu können, muß man vom Programm eine Plotdatei erstellen lassen. Diese wird vom Output-Modul weiterverarbeitet. Die Ausgabe erfolgt über RS 232, Centronics, Midi oder in eine Datei. Für Plotter, Drucker und Laser existieren folgende Treiber:

LASER: SLM 804

DRUCKER: FX 80, FX 100, LQ 850, LQ 2500, P6, P7

PLOTTER: HPGL, MP 3200

Das Ausdrucken bzw. Plotten einer Zeichnung nimmt, je nach Größe und Komplexität, einige Zeit in Anspruch. Deshalb ist es möglich, bis zu 20 Zeichnungen in eine Datei zu schreiben und zu einem geeigneten Zeitpunkt hintereinander auszugeben.Ein Wermutstropfen ist jedoch, daß die ausgedruckten Zeichnungen nicht in allen Punkten dem Original entsprechen. Auf dem Laser SLM 804 und mit dem NEC-Treiber ergaben sich vereinzelt Linienüberhänge, außerdem waren beim Laser einige Linien und die Beschriftung kaum sichtbar. Zudem fehlten am oberen Rand der Zeichnung im Ausdruck einige Millimeter, was sich auf den verschobenen Anfangspunkt zurückführen läßt. Bei der Ausgabe mit dem NEC-Treiber fiel außerdem auf, daß strichpunktierte Linien teilweise als durchgezogene zur Ausgabe kommen.

## CAMPUS CAD 1.3

Wenn man einige Zeit mit CAMPUS DRAFT gearbeitet hat und feststellt, daß man ein umfangreicheres und leistungsfähigeres Programm benötigt, ist die Möglichkeit gegeben auf CAMPUS CAD (siehe Test in Heft 6.87) umzusteigen. Dabei können alle Zeichnungen und Bibliotheken übernommen werden, so daß durch diesen Umstieg keine Verluste entstehen. Dies ist wohl auch der Hintergedanke, der die Veröffentlichung von CAMPUS DRAFT begleitet, denn es ist eine abgemagerte Version, die jedoch Ihre Herkunft nicht verbirgt. Die Benutzeroberfläche und die Bedienung der Funktionen ist weitgehend identisch, doch bietet der große Bruder die Funktionen, die ein professionelles Arbeiten erleichtern.

## CAMPUS DRAFT

Das Programm bietet die für CAD notwendigen Grundfunktionen und leistungsfähige Optionen wie Schraffieren, halbautomatisches Bemaßen und die Symboltechnik. Die Bedienung der meisten Funktionen ist sehr einfach und praxisgerecht. Zwar treten hin und wieder ein paar Unstimmigkeiten auf (siehe z.B. Druckerausgabe) und auch manche Funktion wird man vermissen (z.B. UNDO), doch im großen und ganzen ist das Programm sehr ordentlich gemacht. Bezieht man den Preis in die Überlegungen mit ein, liegt die Zielgruppe für dieses Programm im Bereich der interessierten CAD-Einsteiger und Studenten, die damit zu einem fairen Preis einen Einblick in die vielfältige und komplexe Welt des CAD erhalten können. Für den professionellen Einsatz ist dieses Programm sicherlich nicht gedacht, deshalb sollte man mit den kleinen Unzulänglichkeiten auch nicht hadern, sondern es als gelungenes Einsteigerpaket ansehen und einsetzen.

*MN*

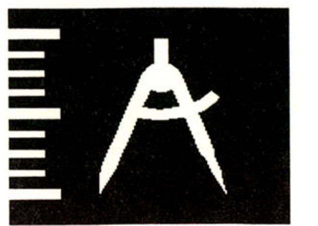
Hersteller: *Technobox Software GmbH*
*Kornharpener Straße 122a*
*4630 Bochum 1*

Preis: *DM 149.-*

Übersicht:
*+ einfache Bedienung*
*+ Symboltechnik*
*+ halbautom. Bemaßung*
*+ Schraffurfunktionen*
*+ 9 Zeichenebenen*
*+ Plotter- u. Lasertreiber*
*+ aufwärtskompatibel zu CAMPUS CAD 1.3*
*- Ausdrucke nicht originalgetreu*
*- Handbuch teilweise ungenau*

Konfiguration:
*mind. 512 KByte und TOS im ROM oder 1 MByte Hauptspeicher Monochrombildschirm*

*ENDE*

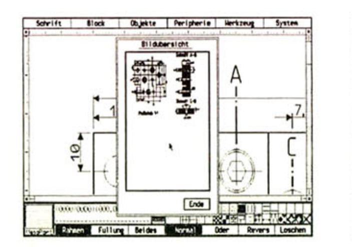


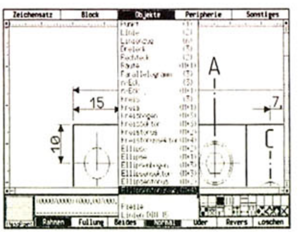
# Der neue Grafik–standard ist da!

## MegaPaint™

Vorbei sind die Zeiten in denen Sie Ihren ST guten Gewissens bestenfalls für Skizzen verwenden konnten. Denn ab jetzt gibt's MegaPaint, das Zeichenprogramm für Profis. Das erste Programm, bei dem gilt: 'What you see is exactly what you get'.

MegaPaint verwandelt Ihren ST in ein Grafikterminal der Superklasse. Da MegaPaint außergewöhnlich hohe Bildauflösungen unterstützt, stellt es eine ideale Ergänzung zum Desktop–Publishing dar. Die Druckqualität der erzeugten Dokumente reicht dabei auch bei weitem für technische Zeichnungen aus.




Die wichtigsten Funktionen:

* Die Bildauflösung orientiert sich an der maximal erzeugbaren Druckauflösung: What you see is exactly what you get.

* Variable Skalierungen und Raster. Formatgetreue Maßstäbe und exakte Positionierung von Zeichenobjekten.

* Spezielle Drucktechnik für Nadeldrucker (NEC P6 und EPSON FX–80) und Atari Laserdrucker (180, 240x216, 300 und 360 Bildpunkte/Zoll) ergeben exakteste DIN A4 Dokumente. Selbst auf einem 8–Nadeldrucker sehen die Dokumente aus, wie auf einem P6 gedruckt!

* Freie Druckeranpassung, Druckertreiber für alle gängigen Fabrikate.

* Vorhandener Text und Grafik kann importiert werden. Kompatibel zu STAD, DEGAS und 1ST Word. Weitere Grafik– und Textformate in Vorbereitung!

* Ideal für Scanbearbeitung geeignet!

* Drehen von Objekten.

* 7 proportionale Fonts nach DIN 6776 Typ B. Fonteditor. Linien nach DIN 15.

* Übersichtlicher, mit der Maus leicht zu bedienender Arbeitsplatz. Bearbeiten einer ganzen A4–Seite per Rollbalken.

* Online–Help und Undo–Funktion. Sichere, speicherplatzsparende Dateiverwaltung.

* Taschenrechner, Lupe und Zoomfunktion

* Unterstützt Blitter

* Kein Kopierschutz

* Ausführliche deutsche Dokumentation (ca. 100 Seiten). MegaPaint ist ein deutsches Produkt und wird mit Upgradegarantie geliefert!

MegaPaint benötigt mindestens 1MB RAM und einen monochromen Bildschirm.

Auf der CeBIT 88 waren sich alle einig: dies ist der neue Standard auf dem ATARI ST!

**Preis: DM 298.–***



## 1ST FREEZER //

Sichern Sie Ihre wertvollen, kopiergeschützten Orginale!

Viren, Hacker und Hardwaredefekte sind unvorhersehbare Gefahren für die teuren Orginalprogramme.

1ST FREEZER™ ist ein Programm, das mit einer völlig neuen Technik arbeitet. Es kopiert nicht die (kopiergeschützte) Diskette, sondern den Inhalt des Hauptspeichers. Damit sind auch bisher nicht kopierbare Disketten sicherbar. Es besteht zusätzlich die Möglichkeit jederzeit das Programm anzuhalten (z.B. Kaffeepause) oder aber eine knifflige Situation erneut zu spielen.

Ein absolutes Muß für jeden Spieler aber auch Büroanwender! Jetzt Version 2 lieferbar. Läuft auch mit Harddisk und MegaST!

(ATARI SPECIAL 3/88: '1ST Freezer ist ein ideales Hilfsmittel.')

**Preis: DM 148.00***

Weiterhin lieferbar:

der Vorgänger 1ST Freezer.

**Preis: DM 114.95***

**Upgrades: DM 30.– und Orginaldiskette.**



# Das Delirium schlägt zurück!

## *StarTrash™*

Der Programmierer von Dizzy Wizard und MusiX32 hat wieder zugeschlagen. Viel Action und Ästhetik sind angesagt. Hier die Story:

Der Weltraum. Unendliche Warteschlangen. Wir stempeln das Jahr 2215. An Bord der Behörde: Captain Flirt, Mr. Spoil, Dr. Jeckel und Mr. Scotch. Auf der Suche nach den verloren gegangenen Aktenordnern der Galaxis.

Die Mission: Wiederbeschaffung aller Akten, die während einer Bordparty versehentlich auf den Planeten StarTrash VII gebeamt wurden. Doch StarTrash VII ist kein



x–beliebiger Planet! Er ist eine konföderierte Besserungsanstalt. Und daß da Captain Flirt & Co. nicht gerne gesehen sind, ist wohl logo. Doch mehr wird nicht verraten!

Ein Spiel gedreht in Dramatovision in den TommySoftware–Studios. Wir sind sicher: das wird ein neuer Klassiker!

**Preis: DM 79.95***

## MULTI ST™

Sensationell! Was bisher nur wesentlich teureren Computern vorbehalten war, wird jetzt auch auf dem ATARI ST möglich: Mehrere Programme gleichzeitig im Rechner.

Vorbei sind die Zeiten, in denen Sie sich in Ihrer Textverarbeitung befanden und z.B. eine Grafik ändern wollten. Dazu mußten Sie Ihr Textverarbeitungsprogramm verlassen, vorher Ihre Daten sichern, warten bis der Schreibtisch aufgebaut war, dann das Grafikprogramm starten, die Grafik laden, Grafik ändern, abspeichern, Programm beenden, Textverarbeitung starten und erneut den Text einladen. Dies läßt sich jetzt mit einem (!) Tastendruck erledigen.

MULTI ST™ erlaubt mit einer speziellen Komprimierungstechnik bis zu zehn(!) Applikationen gleichzeitig in einem 1MB–Rechner. Läuft u.a. mit: DEGAS, SIGNUM, 1ST WORD, STAD, MEGAPAINT u.a.

MULTI ST™ benötigt 1MB RAM, TOS in ROM und läuft in allen Auflösungen. Natürlich arbeitet Multi ST auch problemlos auf dem MEGA ST!

**Preis: DM 148.–***

**Wir vermarkten Ihre ATARI– oder AMIGA–Programme. Schreiben Sie uns!**

**Bestellcoupon**

[ ] Bitte senden Sie mir kostenloses Infomaterial (DM 1,40 in Briefmarken liegen bei) !

[ ] Bitte senden Sie mir 2 Demodisketten (StarTrash & MegaPaint). DM 20.– beiliegend!

[ ] Hiermit bestelle ich: ______________________

zzgl. DM 6.– Versandkosten (bei VK) [ ] per Nachnahme [ ] V–Scheck anbei!

Name:______________ Strasse:______________ Ort:______________ STC 7/88

## 1ST SPEEDER //

Dieses Spitzenprogramm beschleunigt mit modernster Cachetechnik und einem speziellen Precatchverfahren Diskzugriffe durchschnittlich bis zu 1100%, Harddiskzugriffe bis 400%.

Diese Werte sind unerreicht! TurboDos wird als PD mitgeliefert! Autokonfiguration. Bis 16MB–RAM verwaltbar. Optional manuelles Mediachange.

Gemischter Betrieb von Harddisk und Floppy ist möglich. 1ST SPEEDER // läuft auf jeder ST–Konfiguration.

**Preis: DM 114.95 ***

Vorgänger 1ST Speeder

**Preis: DM 89.–***

**Upgrades: DM 30.– und Disk.**

Außerdem:

LISP Complete (98) – MusiX32 (89) – Lock_it (298) – Dizzy Wizard (54.95) – TrashHeap (54.95) – BTX für ST (398) – BTX für Mac (598) – Intelligent Spooler (98) – Adimens (189) – AdiTalk (189) – STAR Drucker NB24–10 (1298)

Alle Programme werden auf einseitigen Disketten geliefert. *Alle Preise sind empfohlene Verkaufspreise! Irrtum und Änderungen vorbehalten.

**WIR SIND UMGEZOGEN!!**

TommySoftware®

Selchower Strasse 32, D–1000 Berlin 44, Tel.: 030/6214063 Q BTX:0306214063. TLX via BTX.

CH: Senn Computer AG, Langstrasse 31, CH–8021 Zürich, Tel.: 01/2417373

A: Computerstudio Wehsner GmbH, Paniglgasse 18–20, 1040 Wien, Tel.: 0222/657808

TEST

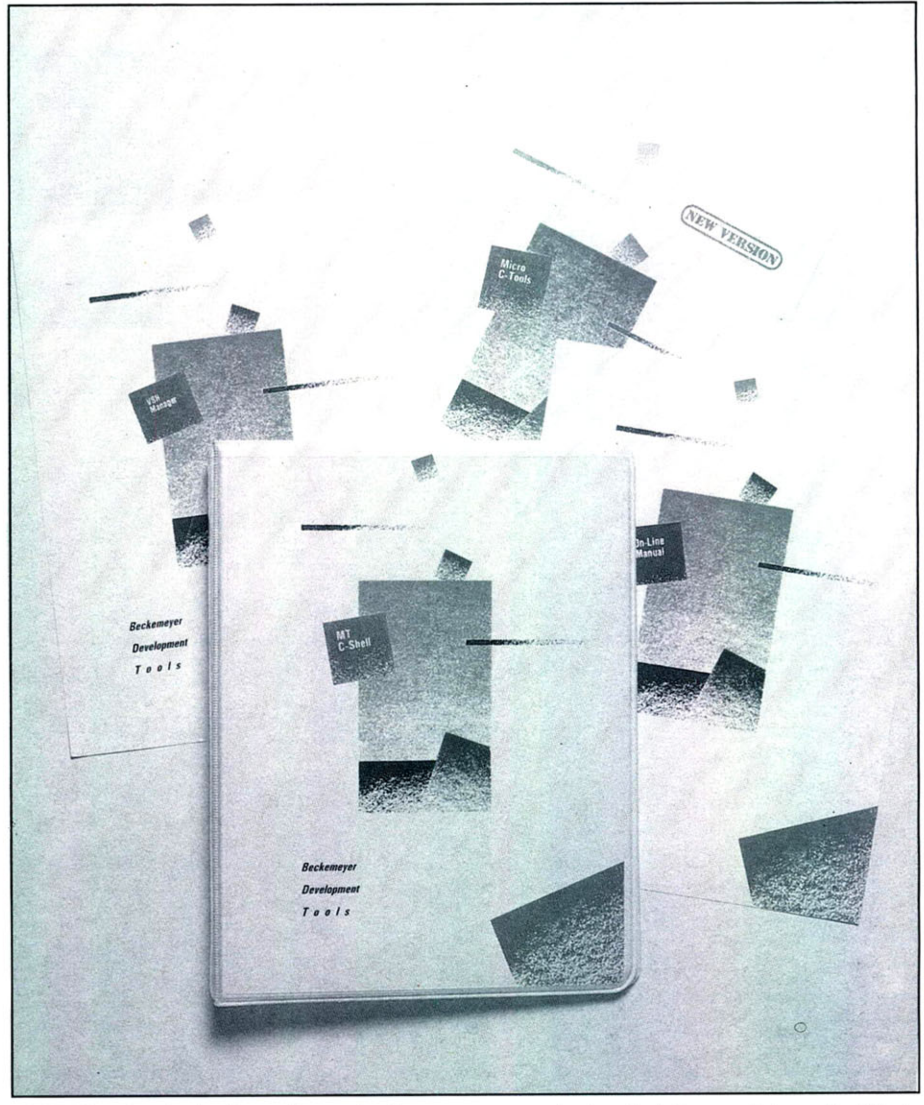
# MULTITASKING AUF DEM ST ?

## und es geht doch !

**Unter dem Namen 'Beckemeyer Tools' sind für den Atari ST einige interessante Tools verfügbar, die besonders für die Softwareentwicklung mit der Sprache C hervorragend geeignet sind.**

Das Software-Paket besteht aus mehreren Einzelprogrammen, die sich mit Hilfe eines Installationsprogrammes einfach auf einer Festplatte installieren lassen und fortan hervorragend miteinander harmonieren.
Folgende Einzelmodule sind verfügbar:
- MT C-Shell (Kommando-Interpreter)
- Micro C-Tools (Befehlserweiterung der Shell)
- VSH-Manager (Betrieb der MT C-Shell unter GEM-Programm und -Accessory)

Gesteuert werden die ganzen Tools über die MT C-Shell, einen Kommandointerpreter, so wie man ihn vom UNIX bzw. MS-DOS- Betriebssystem her kennt. Also nicht gerade das Wahre für alle fanatischen Anhänger von Fenstern, Icons und Plastikmäusen. Doch was sich mit dieser Shell so alles anstellen läßt, sollte zumindest einige dazu anregen, sich von dem grauen Tier loszusagen, und mal so richtig in die Tasten zu hauen. Ein kleiner Wermutstropfen bleibt dennoch für diese Leute übrig. Mit Hilfe des VSH-Managers läßt sich die Shell auch in einem GEM- Window betreiben und mit Hilfe des VSH-Accessory sogar aus jeder GEM-Anwendung, die Accessories zuläßt, starten.
UNIX-Kennern fällt beim ersten Arbeiten mit der Shell sofort die frappierende Ähnlichkeit mit diesem Betriebssystem auf. Dadurch ist auch die enge Verbundenheit mit der Sprache C zu erklären, die ja gewissermaßen Muttersprache von




UNIX ist. So sind neben allen unbedingt nötigen Befehlen zum Filehandling (copy, rename etc..) auch Befehle zum Steuern des C-Compilers (hier der vom DRI- Entwicklungssystem) sowie zum Vergleichen, Untersuchen, Durchsuchen von Files mit bzw. nach Zeichenketten, sowie Kommandos zur Benutzung von Jobs, Batchfiles und Pipes vorhanden. Laut Aussage von Benutzern werden sogar in der Anleitung nicht aufgeführte UNIX-Befehle anstandslos ausgeführt. Am Schluß dieses Artikels ist eine Auflistung der Befehle abgedruckt.
Durch die Multitasking Fähigkeiten lassen sich Jobs wie Drucken, Diskformatieren oder Compilierdurchläufe im Hintergrund erledigen, während man etwa einen Text bearbeitet. Hier eröffnen sich für den Atari ST-Anwender ganz neue

Perspektiven. Im Gegensatz zum RTOS oder OS/9 (beides sind Multitasking-Betriebssystem für den ST) lassen sich bei der der MT C-Shell ganz normale TOS oder auch GEM-Applikationen starten. Bei Verwendung des VSH-Manager kann man Parallel-Prozesse in verschiedenen GEM-Fenstern darstellen. Leider funktioniert dies nicht mit GEM-Anwendungen.
Eins sollte jedoch nicht unerwähnt bleiben: durch die Kompatibilität zu TOS- und GEM- Applikationen leidet die Ausführungsgeschwindigkeit der Programme, besonders bei der Verwendung des VSH-Managers, doch erheblich.

**Hardware Voraussetzung:**

- Atari ST Computer mit min. 1 MByte RAM
- Doppelseitige Diskstation (Auf Wunsch wird die Software auch auf einseitigen Disketten geliefert) und
- Festplatte.

**Positves:**

- Multitasking, multiuserfähig mit Passwortschutz
- Befehlssyntax kompatibel zum UNIX-Betriebssystem
- Modulares Konzept, dadurch leicht erweiterbar

**Negatives:**

- englisches (Referenz-)Handbuch, nur zum Nachschlagen geeignet, keine Einführung und keine Beispiele zum UNIX-kompatiblen Befehlssatz.
- Langsame Textausgabe, besonders unter GEM und bei Multitaskingaufgaben

**Preise:**

| | |
|---|---|
| MT C-Shell: | 298,- DM |
| Micro C-Tools: | 79,- DM |
| VSH-Manager: | 119,- DM |
| UUCP: | 139,- DM |
| Development System: | 159,- DM |

## Fazit

Für all diejenigen, die mit UNIX zu tun haben und mit der Entwicklung von Software beschäfftigt sind, dürfte sich die Anschaffung dieser Shell lohnen, besonders wenn man bedenkt, was man alles an Werkzeugen mit auf den Weg bekommt. Diese Gruppe von Anwendern wird sich dann auch nicht über das wirklich magere Handbuch, das zudem noch in Englisch abgefaßt ist, ärgern. Ist doch die Anschaffung weiterführender Literatur über UNIX ein absolutes Muß, um ein vernünftiges Arbeiten mit der Shell zu gewährleisten. Doch einmal mit dieser Literatur "bewaffnet", ist die Shell auch für UNIX-Neulinge zum Einstieg in die faszinierende Welt der Großrechenanlagenbetriebssysteme (Puh, was für ein Wort!) denkbar.
Der Distributor kündigte eine neue Version der Shell sowie zusätzliche Tools an (Erscheinungstermin Juni '88). Dazu gehört u.a. das UUCP (UNIX-to-UNIX-Copy). Es handelt sich dabei um ein weltweites Kommunikationsnetz von UNIX-Rechnern, die sich gegenseitig auf freiwilliger Basis Nachrichten übermitteln können. Ferner ist noch ein Software Development System für die MT C-Shell erhältlich, in dem u.a. wichtige Funktionen als C-Bibliothek (mit Source-Code) enthalten sind.

*H. J. Reiss*

*ENDE*

# 3.0




GFA-BASIC – ohnehin eine der leistungsstärksten BASIC-Versionen für den ST – ist jetzt in der neuen, von allen ST-Anwendern sehnlichst erwarteten Version 3.0 erhältlich. Neue Strukturanweisungen, Folding-Technik, zahlreiche Schleifenbefehle und eine höhere Geschwindigkeit ermöglichen nun eine noch effektivere Programmierung. Eine ideale Einführung in den gesamten Leistungsumfang dieses Interpreters und seiner Vorgänger findet der Anfänger in „GFA für Einsteiger". Vom einfachen Start bis hin zum ersten GEM-Projekt wird hier alles ausführlich und anhand praktischer Beispiele erläutert: Sprünge und Schleifen, Daten und Operationen, Prozeduren und Funktionen mit Parametern, Daten-Bearbeitung, Diskettenzugriff, Grafik und, und, und. Dazu die nötigen Tips für die Anpassung von GFA-2.0- auf GFA-3.0-Programme! GFA für Einsteiger hat ca. 250 Seiten und wird ab ca. 7/88 für DM 29,– erhältlich sein.
Wer noch tiefer in die professionelle Programmierung mit GFA-BASIC vorstoßen will, sollte mit dem großen GFA-BASIC-Buch arbeiten. Dieses Buch verleiht Ihnen den letzten Schliff mit all den nötigen Tips und Tricks. Anhand einer RAM-internen Dateiverwaltung lernen Sie diese mächtige Interpreter-Sprache professionell zu nutzen. And-Or-Verknüpfungen, mehrzeilige Suchvorgabe, Sortieralgorithmus, Nutzung der AES-Routinen zur Window- und Objektprogrammierung – alles, was Sie für eine professionelle Programmentwicklung unter GFA brauchen, können Sie hier praxisnah lernen und anwenden. Natürlich auch in diesem Buch eine ausführliche Beschreibung des GFA-Compilers und eine komplette Befehlsübersicht der Version 3.0! Das große GFA-BASIC-Buch – in der Neuauflage mit knapp 600 Seiten ab ca. 7/88 für DM 49,– erhältlich.
Nun könnte man meinen, wer mit dem großen GFA-BASIC-Buch gearbeitet hat, ist in Sachen GFA-BASIC absolut fit. Stimmt auch! Doch warum das Rad nochmal erfinden? Das große GFA-Programmierhandbuch bietet Ihnen für die eigentliche Programmentwicklung Lösungen, die Sie für Ihr Programm sofort verwenden können. So finden Sie hier die wichtigsten Standardalgorithmen aus nahezu allen Bereichen. Ob Sortier-Routinen oder Verketten von Feldern, ob bestimmte Algorithmen zur Textverarbeitung oder Dateiverwaltung, zu Grafikprogrammen oder zu mathematischen Berechnungen, auf über 400 Seiten zeigt Ihnen dieses Buch, wie Sie solche Probleme richtig angehen. Auf einer mitgelieferten Diskette finden Sie zusätzlich noch eine Reihe nützlicher Tools wie einen REM-Killer für GFA-BASIC oder ein Tool für ein automatisches Backup-System. Für DM 59,– können Sie sich somit jede Menge Ärger beim Programmieren sparen.

**DATA BECKER**
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

EINSENDEN AN: DATA BECKER · MEROWINGERSTR. 30 · 4000 DÜSSELDORF 1
Hiermit bestelle ich für meinen ATARI ST
Name, Vorname
Straße
Ort
zzgl. DM 5,– Versandkosten unabhängig von der best. Stückzahl




# KatCe - ST

## - ein Pascal- und Assembler-Entwicklungssystem

**Der Name dieses Softwarepakets wird im einen oder anderen Leser Erinnerungen wecken. Richtig! Genau das war es, da gab es doch - Verzeihung, gibt es einen MC68000-Einplatinen-Computer mit dem gleichen Namen. Kenner der Szene wissen, daß es dafür auch einen Pascal-Compiler gibt. Sollten da etwa Ähnlichkeiten vorhanden sein? Die Antwort ist ja. Es handelt sich um eine Portierung des Entwicklungssystems der Kat 68-Systeme aus der Pionierzeit des 68000-Prozessors. Damals war das noch Hi-Tech (neueste Technologie). Heute in der Zeit der 68030er und Transputer nimmt ein Rechner, der *nur* einen 68000 hat, eine ähnliche Stellung ein wie der C-64 zu Anfang der Atari ST-Zeit.**

Aber nun zur Software. Was bekommt man für die 100 Märker, die KatCe-ST kostet, denn geboten? Erstmal einen Pascal-Compiler, der in puncto Geschwindigkeit alles andere zersägt. Dazu gibt's einen Editor, der sehr stark mit Wordstar/Turbo-Pascal verwandt ist. Wer gerne Bits fummelt, freut sich sicher, daß außerdem noch Assembler, Disassembler und Debugger dazugehören. Das ganze ist in ca. 83 Kbyte zusammen mit einer Shell, die an das UCSD-System erinnert, untergebracht. Die Größe läßt darauf schließen, daß bei der Entwicklung keine Hochsprache, sondern reine Maschinensprache verwendet wurde. Ich werde jetzt zu jeder einzelnen Komponente des Systems ein paar Worte verlieren. Beginnen wir mit dem

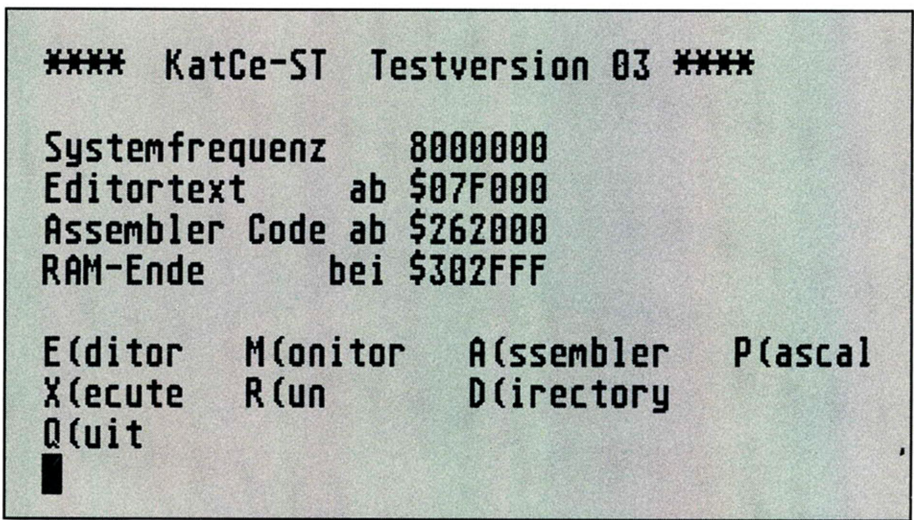



*Bild 1: So meldet sich KatCe-Pascal*

## Pascal-Compiler

Wie bei den meisten anderen Compilern auch, wird zusätzlich zum Standard ein Sack voll Erweiterungen geboten. Das freut den Kunden, macht sich gut in Werbebroschüren und Anzeigen und schränkt die Portabilität ein, weil natürlich kein anderes System die gleichen Erweiterungen bietet. Das wiederum bindet den Kunden an den Hersteller und der freut sich. So sind alle zufrieden, oder?

Genug des Unkens, gehen wir zusammen die Liste durch. Das Vorhandensein des Assemblers ist natürlich ein Grund, eine entsprechende Schnittstelle zu bieten. Es besteht die Möglichkeit, Prozeduren, die in Assembler geschrieben sind, aufzurufen. Eine für Pascal eher exotische Erweiterung sind parallele Prozesse. Bis zu 8 Tasks können parallel laufen und über globale Variablen Nachrichten austauschen. Genauso ungewöhnlich ist es, daß der Editor als Funktion von Pascal aus aufgerufen werden kann; doch dazu später mehr. Wie es sich für den ST gehört, sind natürlich die AES- und VDI-Funktionen im Sprachumfang enthalten. Soviel zu den Erweiterungen.

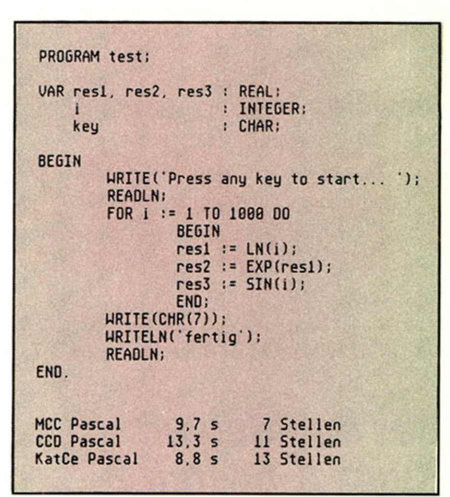
Wie ich vorhin schon erwähnte, ist der Compiler sehr schnell. Um konkret zu sein: er verarbeitet 21.500 Zeilen pro Minute. Im Handbuch wird die Übersetzungsgeschwindigkeit mit 200 Zeilen pro Sekunde angegeben (wieviel sind das nochmal pro Minute? Stimmt! 12.000). Der Hersteller ist an dieser Stelle erstaunlich bescheiden. Ich habe jedenfalls den Wert von 21.500 anhand des mitgelieferten Programms DEMO.PAS ermittelt.

Woher kommt die auffallend hohe Geschwindigkeit, werden Sie sich fragen. Ganz einfach, alles läuft im Speicher. Der Compiler und der Quelltext befinden sich immer im Speicher und das erzeugte Programm wird auch dort abgelegt. Wie bei Turbo-Pascal kann man optional auch mit Dateien auf Floppy oder Harddisk arbeiten. Dann dauert's natürlich länger. Der erzeugte Code kann sich auch sehen lassen. Besonders erwähnenswert ist die Real-Mathematik. Der Rechenbereich

reicht von 10 hoch -999 bis 10 hoch +999 mit einer Genauigkeit von 13 Stellen (CCD-Pascal bietet 11 Stellen). Faul, wie der Mensch nun einmal ist, habe ich als Benchmark ein altes Programm hergenommen, das ich schon in der ST Nr.6 anno 1986 verwendet habe, um den Pascal-Compiler von Metacomco zu testen. Ja, ich weiß, sie haben Ihren ST noch nicht so lange, oder Sie haben keine Lust, in den Keller zu gehen und in der Kiste mit den alten Zeitschriften zu wühlen. Keine Aufregung, das Listing finden Sie in Bild 2, da stehen auch die Ergebnisse des Zeitvergleichs. Natürlich habe ich auch den obligatorischen Test mit dem Sieb des Eratosthenes durchgeführt (den drucke ich aber nicht zum 1001. Mal ab). Der Prüfling benötigte 25,6 Sekunden für 25 Iterationen, Pascal 2.02 von CCD 15,1 Sekunden. Borland wirbt mit 9,3 Sekunden für Turbo-Pascal auf einem 8 MHz-AT. Aber denken Sie daran, daß Primzahlen nicht alles sind.
Erinnern Sie sich noch daran, daß man von Pascal aus den Editor aufrufen kann? An dieser Stelle mehr dazu. Die Entwicklungsumgebung enthält außer dem Editor noch jede Menge anderer Funktionen, die von mehreren Stellen aus gemeinsam verwendet werden. Viele dieser Funktionen kann man auch von Pascal aus aufrufen. Das bedeutet aber, daß die Routinen auch bei einem eigenständigen Programm (neudeutsch: stand-alone-Application) vorhanden sein müssen. Es muß also zu jedem noch so kleinen Programm ein Laufzeitmodul mit einem Umfang von ca. 25 KByte gebunden werden, um es vom Desktop aus starten zu können.
Der Compiler unterstützt Include-Dateien, die aber nicht geschachtelt werden können. Ansonsten gibt es keine Modularisierung, es gibt ja auch keinen Linker. Nun zum

## Assembler

Der 2-Pass-Assembler versteht die Motorola Mnemonics und kann sowohl Dateien includieren als auch verkettete Quelltexte verarbeiten. Der Pascal-Compiler wurde laut Aussage des Entwicklers



```
PROGRAM test;

VAR res1, res2, res3 : REAL;
    i                : INTEGER;
    key              : CHAR;

BEGIN
      WRITE('Press any key to start... ');
      READLN;
      FOR i := 1 TO 1000 DO
            BEGIN
            res1 := LN(i);
            res2 := EXP(res1);
            res3 := SIN(i);
            END;
      WRITE(CHR(7));
      WRITELN('fertig');
      READLN;
END.

MCC Pascal      9,7 s    7 Stellen
CCD Pascal     13,3 s   11 Stellen
KatCe Pascal    8,8 s   13 Stellen
```

*Bild 2: Testroutine und Ergebniss*

mit diesem Assembler erstellt. Einen gründlicheren Test kan man sich kaum vorstellen.
Mit dem Assembler kann man auch Programme erstellen, die *nicht* das Laufzeitmodul brauchen. Allerdings muß man dann auch auf die Annehmlichkeiten der darin enthaltenen Funktionen verzichten. Mehr ist zu dem Assembler nicht zu sagen. Machen wir weiter mit dem

## Editor

Wie gesagt ist der Editor von den Kommandos her Turbo-Pascal kompatibel. Er verarbeitet allerdings keine Standard-Textdateien wie sie auf dem ST üblich sind, sondern hat ein eigenes Dateiformat. Umwandlungsroutinen von und nach ASCII werden in Pascal mitgeliefert. Die Konvertierroutinen lesen und schreiben immer in 512 Byte-großen Blöcken. Dadurch kommt es in der Regel zu Fehlermeldungen (welche Datei hat als Größe schon genau ein Vielfaches von 512 Byte ?). Aber dafür hat man ja den Quelltext und kann basteln. Ich gehe davon aus, daß der Autor, wenn Sie diesen Artikel lesen, die Konvertierung schon überarbeitet hat. Als letztes noch ein paar Bemerkungen zum

## Debugger

Man kann damit den Speicher ansehen und modifizieren. Speicherbereiche können in eine Datei gespeichert und von einer Datei gelesen werden. Es lassen sich Befehle im Einzelschrittmodus abarbeiten aber keine Unterbrechungspunkte (Breakpoints) setzen. An dieser Stelle noch ein Nachtrag zu den Erweiterungen des Pascal-Compilers. Es gibt eine Break-Funktion, die es ermöglicht, vom laufenden Programm in den Debugger zu verzweigen und sich die Variablen des Pascalprogramms anzusehen. Dieses Feature ist bei der Fehlersuche sehr hilfreich, es setzt jedoch Assemblerkenntnisse voraus.

## Das Handbuch

ist in deutscher Sprache geschrieben und 180 Seiten lang. Man findet sich relativ schnell zurecht, woran der Suchindex nicht unschuldig ist. Der Schreibstil ist sachlich und knapp, es werden alle Funktionen erklärt, an manchen Stellen wäre jedoch etwas mehr Ausführlichkeit angebracht.

## Fazit

Es bleibt dem Benutzer nicht verborgen, daß dieses Entwicklungspaket aus der Welt der Einplatinencomputer kommt. Die Benutzeroberfläche läuft nicht unter GEM, ist aber akzeptabel. Das exotische Datenformat des Editors ist störend, kann aber hingenommem werden. Auf jeden Fall stimmt das Preis-Leistungs-Verhältnis. Für 100 DM bekommt man sonst nirgends ein Paket mit diesem Funktionsumfang. Softwarefirmen werden für die Entwicklung professioneller Produkte sicher leistungsfähigere (und teurere) Systeme verwenden, aber für den Anwender mit etwas schmalerem Geldbeutel ist KatCe-Pascal eine Alternative.

*JL*

*Bezugsquelle:*
*Christa Mayer-Gürr*
*Software und Computerbaugruppen*
*Treptower Str. 2*
*4350 Recklinghausen*

*ENDE*

TEST

# GETEILTE FREUDE....

**Unter diesem Titel haben wir im vergangenen Jahr das kleine Utility-Programm K-Switch getestet, mit dem es möglich ist, auf einem ST mehrere Programme gleichzeitig im Speicher zu halten und mit einer einfachen Tastenkombination zwischen diesen Anwendungen umzuschalten. Jetzt gibt es gleich drei neue Programme dieser Art. Dieser Test zeigt, ob sie halten, was sie versprechen. Also Vorhang auf für Kumas K-Switch 2, Steinbergs Key-Switcher und Tommy Softwares Multi ST.**

Wofür braucht man eigentlich sowas? Beispiel: Sie schreiben einen Text und benötigen Illustrationen. Also greifen Sie zur Software-Säge und teilen Ihren Rechner in zwei Hälften: in die eine laden Sie das Text- und in die andere ein Malprogramm. Jetzt brauchen Sie nicht mehr dauernd ihre Arbeit abzuspeichern, das eine Programm zu verlassen, das nächste Programm einzuladen usw., usw., sondern nur noch schnell zwischen den beiden Programmen umzuschalten. Ein paar Tasten und Sie sind in einem völlig anderen Rechner. Ist das nicht erfreulich? Geteilte Freude ist eben doch doppelte Freude.

Das alte K-Switch-Programm konnte den Rechner nur in zwei Hälften teilen. Wenn Sie aber einen Rechner mit mehr als einem Megabyte Speicher haben, kann es auch sinnvoll sein, den Speicher auf mehr als zwei Programme zu verteilen. Die neueren Programme sind da schon etwas vielseitiger. Der Reihe nach sollen sie nun vorgestellt werden, auf das Kompatibilitätsproblem komme ich dann am Ende für alle drei Programme gemeinsam zu sprechen:

## Der Nachfolger: Kuma K-Switch 2

K-Switch 2 ist in der Lage, den Rechner in bis zu fünf Teile zu zerlegen. Wie beim alten K-Switch ist dabei die Größe der einzelnen Teile nicht wählbar. Wenn Sie z.B. vier Megabyte Speicher haben und Sie wollen vier Teile, ist eben jedes knapp 1 Megabyte groß, wenn Sie nur zwei Teile benötigen, wird jedes 2 Megabyte Speicher verwenden.

K-Switch schaltet nicht nur zwischen Programmen um, sondern es erzeugt für jeden Teil einen komplett neuen Desktop, den Sie vollkommen unabhängig von den anderen Desktops aufräumen und verwalten können. Selbst die Maus wird beim Umschalten zwischen zwei Teilen so verschoben, daß sie immer dort steht, wo sie beim Verlassen des Teiles zuletzt gewesen ist.

Für den Benutzer sieht es tatsächlich so aus, als würde er per Tastendruck auf einen ganz anderen Rechner umschalten. Ein Beispiel für eine Anwendung: Die vielbeschworene Kombination aus Text- und Malprogramm und zusätzlich noch ein Desktop zum Kopieren von Files, oder um kurz eine andere Anwendung 'dazwischenzuschieben'.

Die Bedienung von K-Switch 2 ist denkbar einfach. Man startet das Programm per Doppelklick und muß dann nur noch die Frage beantworten (in einer Dialogbox), in wieviele Teile man den Rechner zerlegen will. Die Dialogbox verschwindet und man findet sich auf dem Desktop wieder. Das war's. Umschalten kann man nun mit der Tastenkombination <Linke Shifttaste / Rechte Shifttaste / Alternate>. Dabei kann man die gewünschte Ziel-'Hälfte' nicht direkt anwählen, sondern muß immer der Reihe nach durch alle Teile durch: Von 1 nach 2 nach 3 nach 4 nach 5 nach 1 usw.. Das hat den Vorteil, daß nur eine einzige Tastenkombination benötigt wird, aber auch den Nachteil, daß man eben schlimmstenfalls durch vier Teile durchmuß, die einen überhaupt nicht interessieren. Leider braucht das Umschalten selbst auch so seine Zeit: Zwischen 1 und 2 Sekunden dauert es, bis der 'neue' ST erscheint.

Man kann seinen Rechner ohne auszuschalten wieder zusammensetzen, indem man das K-Switch-Programm noch einmal startet. Danach ist man wieder auf dem Desktop, jedoch diesmal auf dem einzigen, dem wahren Desktop. Natürlich geht dabei alles, was eventuell noch in den anderen Rechnerteilen gespeichert war, verloren.

Die Anleitung zu K-Switch 2 ist kurz und schmerzlos, in Anbetracht der einfachen Bedienung ist mehr aber auch wirklich nicht erforderlich. Die resetfeste Ramdisk K-Ram wird übrigens mitgeliefert.

Ärgerlich ist es, daß man die Größe der Teile nicht einstellen kann. Wenn man nur ein kleines Utility im Hintergrund haben möchte, braucht es genausoviel Speicherplatz, wie die große, große Datenbank, mit der man eigentlich arbeitet. Das reduziert den Wert eines Switcher-Programmes für Anwendungen, bei denen man nicht zwischen großen Anwendungen, sondern zwischen einem großen und ein paar kleinen Programmen wechseln will, ganz erheblich.

## Der Switcher mit dem besonderen etwas: Steinberg Keyswitch

Steinberg in Hamburg ist ein Musiksoftwarehaus. Das merkt man auch dem verhältnismäßig kleinen Produkt, um das es in diesem Test geht, an, obwohl es durchaus auch völlig unmusikalisch nutzbar ist.

Der Steinberg-Switcher arbeitet grundsätzlich anders, als K-Switch 2. Das Programm teilt den Rechner nicht in bis zu fünf eigenständige Desktops auf, sondern erlaubt es, bis zu 10 Programme auszuwählen und zu starten, die dann über das Keyswitch-Programm umgeschaltet werden können.

Dazu erscheint nach dem Start eine Dialogbox, die der Konfigurierung dient. Für jedes der 10 Programme kann angegeben werden, wieviel Speicher es benötigt, sowie der Pfad, auf dem es auf Diskette oder Harddisc (mittels Fileselectorbox) zu finden ist. Außerdem sind noch ein paar Spezialitäten für Musiker eingebaut, doch dazu später mehr.

Wenn man die Programme von Diskette laden will, sollte man das sofort tun, denn bei einem späteren Start muß immer wieder die richtige Diskette im Laufwerk sein (ein Programm wird eben erst dann geladen, wenn man das erste Mal auf dieses Programm umschalten will). Wenn das Programm aber einmal im Speicher ist, wird die Diskette nicht mehr gebraucht.

Die ganze Konfiguration läßt sich natürlich auf Diskette abspeichern und wird beim Starten des Programmes automatisch geladen. Jedes Programm wird dann, wie bereits erwähnt, beim ersten Aufruf geladen. So verbringt man keine unnütze Zeit mit Warten auf ein Programm, das bei der momentanen Arbeit vielleicht gar nicht benötigt wird.

Wenn KeySwitch einmal läuft, gibt es keine Möglichkeit mehr, auf den Desktop zu gelangen. Anders als bei K-Switch ist eben nicht der Desktop die Basis für den Umschaltvorgang, sondern das Key-Switch-Programm selber. Aus einer der 10 Anwendungen schaltet man mit <Control / Linke & Rechte Shifttaste> auf die Keyswitch-Dialogbox, aus der man mit der Maus (durch Anklicken eines der zehn Programme, die in der Dialogbox eingetragen sind) oder mit den Funktionstasten dann ein anderes Programm erreichen kann. Das Umschalten geht schnell, allerdings erfolgt die Reaktion auf die Umschalttasten nicht immer prompt: Erst, wenn die nächste GEM-Funktion in der Anwendung ausgelöst wird, merkt KeySwitch, daß es Zeit fürs Schalten ist. Daraus kann man bereits entnehmen, daß TOS-Anwendungen zwar laufen, aber nicht mehr (außer durch Quit) verlassen werden können. Dies ist übrigens immer die zweite Möglichkeit, wieder zur Switcher-Dialogbox zu kommen: durch Verlassen der jeweiligen Anwendung. Will man sie dann aber wieder aufrufen, muß sie auch wieder von Diskette geladen werden.

Wie K-Switch kann man auch nach Verwendung von KeySwitch zur normalen Arbeitsweise zurückkehren: Es gibt eine 'Quit'-Option, die allerdings nur dann aufgerufen werden kann, wenn alle gestarteten Applikationen ordnungsgemäß beendet wurden.

Nun aber zu den musikalischen Fähigkeiten von KeySwitch. Bei Anwahl eines der zehn Programme kann automatisch ein Midi-Program-Change-Befehl gesendet werden, der angeschlossene Midi-Geräte auf irgendeine Art neu konfiguriert. Gedacht ist das so: Sie haben ein paar Musikprogramme im Rechner, einen Sequencer und ein paar Editoren für ihre Synthesizer. Dummerweise muß das Midi-Setup für die Arbeit mit einem Sequencer meist ganz anders verschaltet sein als für die Arbeit mit einem Editor. Da heißt es Kabel umstecken. Es sei denn, Sie haben eine Midi-Matrix, das ist ein elektronisches Gerät zur Verschaltung von Midi-Leitungen, die sich über Programm-Change-Befehle programmieren läßt. Diese Midi-Matrix könnte Ihr ganzes Setup jetzt beim Umschalten vom Sequencer in einen Editor komplett umkonfigurieren. Es lassen sich aber sicher auch noch andere Anwendungen dafür finden.

Bei den meisten Musikprogrammen nützte das alles aber noch gar nichts, weil diese Programme fast alle durch sogenannte Key-Module, die in den Romport gesteckt werden, kopiergeschützt sind. Glücklicherweise gibt es da aber eine kleine Hardware-Erweiterung von Steinberg, die es den Programmen erlaubt, automatisch die Kopierschutzstecker umzuschalten. Damit ist dann das lästige Umstöpseln total vom Tisch.

## Der dritte im Bunde: Multi ST

Das dritte Switcher-Programm von der Firma Tommy Software nähert sich wieder mehr dem Prinzip des Kuma-Switchers: Auch hier wird der ST in mehrere, maximal 10 'Desktops' aufgeteilt. Allerdings wird der Switcher anders konfiguriert. Die Programmdiskette des Multi ST enthält nämlich mehrere Programmversionen für die verschiedenen ST-Konfigurationen, aus denen man sich eine Version für den eigenen Aufbau auf Harddisk oder Diskette zusammenkopieren muß.

Wenn man dies getan hat, fragt Multi ST den künftigen Benutzer nach der Größe des Speichers pro Desktop (nicht für jeden Desktop einzeln, alle Desktops sind gleich groß) sowie nach der gewünschten Umschalttaste, wobei <Shift>, <Control> oder <Alternate> zur Verfügung stehen. Hat man die gewünschten Werte eingegeben, gibt es erstmal einen RESET. Danach meldet sich der erste Desktop.

Nach dem RESET wird selbstverständlich neu von einer eventuellen Platte oder von Diskette gebootet. Einen neuen Desktop kann man jetzt erzeugen, indem man die vorher ausgewählte Umschalttaste und dann eine der zehn Funktionstasten (oder besser neun Funktionstasten, denn den Desktop 1 gibt es ja schon) drückt. Wieder gibt es einen RESET, und nach dem darauffolgenden Booten befindet man sich auf einem neuen, jungfräulichen Desktop. Auf einen bereits existierenden Desktop kann man umschalten, indem man die Umschalttaste und die entsprechende Funktionstaste drückt; ist die Nummer schon belegt, gibt es keinen Reset, sondern nach ca. 2 Sekunden schaltet Multi-ST mit einem scheußlich aussehenden Bild auf den entsprechenden Desktop (bzw. die von dort gestartete Applikation) um.

Für jeden neu erzeugten Desktop wird neu gebootet. Das bedeutet, daß man eventuell im Auto-Ordner befindliche Hardiskcaches oder Ramdisks besser abschalten sollte, weil sie sonst ganz offensichtlich für jeden Desktop neu eingerichtet werden - mit all ihrem Speicherbedarf natürlich.

Die Anleitung zu Multi ST ist zwar ausführlich, aber nicht immer gut verständlich.

Leider kann man auch bei Multi ST wie bei K-Switch 2 nicht die Größe jedes einzelnen Desktops festlegen. Immerhin ist aber die Anzahl der Desktops flexibler, da man die Größe pro Desktop ja zu Beginn festlegen kann. Völlig entfernen läßt sich Multi ST übrigens nur durch Ausschalten des Rechners. Nach einem RESET kann man sich aber nicht darauf verlassen, daß noch alles unversehrt in den einzelnen Desktops vorhanden ist.

## Wie gut funktioniert's???

Jedes der drei Switcher-Programme war bei mir einige Zeit im praktischen Einsatz. Dabei zeigte sich bei allen dreien, daß die Betriebssicherheit bei installiertem Switcher doch etwas abnimmt. Jede Softwaresäge hat dabei so ihre eigenen Probleme.

K-Switch arbeitet sehr zuverlässig, solange man nicht zuviele Programme über Auto-Ordner startet. Mit einem Harddisk-Cache gab es nie Probleme, aber

solche Dinge wie neue Fileselectorboxen und automatische Abspeicherungen innerhalb von 1st Word lassen die Zuverlässigkeit doch stark sinken. Es gibt aber sonst anscheinend keine bestimmte Art von Anwendungen, die K-Switch verabscheut. GEM- und TOS-Programme arbeiteten jedenfalls in meinen Tests weitgehend zufriedenstellend.

Der KeySwitcher von Steinberg ist konzeptbedingt etwas anspruchsvoller. Da er den Rechner nicht in Desktops aufteilt, auf denen dann beinahe passieren kann, was will, sondern die Verteilung der Software selbst übernimmt, ist die Anwendung auf GEM-Programme beschränkt. Selbst GEM-Software wie Signum, die erst den Start eines Install-Programms erfordert, funktioniert nicht. Accessories funktionieren.

Zwei sehr große Vorteile hat der KeySwitcher: Die Verteilung des Speichers ist einstellbar. Und: es ist der einzige Switcher, der wenigstens mit einigen Midi-Musikprogrammen (zumindest allen von Steinberg) zurechtkommt (dabei muß natürlich erwähnt werden, daß die Musiksoftwarehäuser Dr.T's und Hybrid Arts ihre eigenen Switcher-Systeme haben, die aber im Gegensatz zu KeySwitch ausschließlich mit Programmen der genannten Hersteller arbeiten und nicht für den universellen Einsatz geeignet sind).

Auch Multi ST arbeitet nicht immer problemlos. Auf Rechnern mit wenig Speicherplatz ist sein Einsatz immer dann kritisch, wenn Caches, Druckertreiber oder andere Utilities aus Auto-Ordnern gebootet werden sollen, da diese kleinen Helfer dann sehr viel Speicher schlucken. Auch die Installation und Bedienung ist nicht gerade superelegant. Mich persönlich stört die Optik des Umschaltens, aber das ist eine reine Ästhetik-Frage. Ein paar Probleme gibt es mit manchen TOS-Programmen, die nicht immer so wollen, wie sie sollen. Im Betrieb mit GEM-Anwendungen ist Multi ST aber zuverlässig.

Sie sehen, jedes der Programme hat so seine Macken. In Tabelle 1 sehen Sie noch einmal ein paar Ergebnisse mit wichtigen Programmen zusammengefaßt. Interessant ist, daß GFA-Basic mit keinem der Switcher zufriedenstellend läuft. Eingegebene Programme funktionieren mit K-Switch oder dem KeySwitcher zwar, aber das Editor-Menü ist in der Horizontalen um ein Bildschirmdrittel oder gar eine ganz Bildschirmhälfte verschoben. Recht unkomfortabel. Mit Multi ST ist der Bildschirm zwar nicht verschoben, aber der Editor geht nicht. Geladene Programme kann man aber laufen lassen.

| Programm | K-Switch | Keyswitch | Multi ST |
|---|---|---|---|
| 1st_Word | + | + | + |
| Degas El. | + | + | + |
| STAD | + | + - 1) | + |
| H_D_U | + | + | + |
| Signum | + | - | + |
| GFA-Basic | - 2) | 2) | 2) |
| Publishing Partner | + | + | + |
| TOS-Prg's | + | - | + - |
| Accessor. | + | + | + |

1) Beim Verlassen von STAD gibt es öfter Probleme
2) Die Probleme mit GFA wurden im Text erwähnt

*Tab. 1: Testergebnisse der Switcher Programme*

## Zusammenfassung

Wenn Sie einen Switcher für eine bestimmte Anwendung suchen, sollten Sie vorher unbedingt sorgfältig testen, ob er sich mit den entsprechenden Programmen auch verträgt. Eine Funktionsgarantie mit allen Programmen gibt daher auch keiner der Hersteller.

Konzeptbedingt ist der Einsatzbereich des Steinberg Switchers etwas eingeschränkt auf GEM-Programme, dafür ist er am anpassungsfähigsten; keines der beiden anderen Programme erlaubt es, die Rechneraufteilung an den Speicherbedarf der Anwendungen anzupassen. Für Musik-Anwendungen ist der Steinberg-Switcher sogar prädestiniert. Auch Programme, die ein Installationsprogramm benötigen (z.B. Signum), sind nicht geeignet, mit dem KeySwitch betrieben zu werden. Nur eigenständige GEM-Programme funktionieren, aber dafür mit Komfort und automatischer Konfiguration mit Hilfe einer Setup-Datei. Leider ist die Umschaltung nicht ganz so sauber gelöst wie zum Beispiel bei K-Switch, man muß nach Betätigen der Umschalttasten immer erst eine GEM-Funktion aufrufen, bevor der Switcher merkt, daß er gemeint ist. Manchmal nervt das, aber man wird ja nicht pausenlos umschalten...

K-Switch ist einfach im Handling, allerdings sind Programmwechsel etwas umständlich, wenn mehr als zwei 'Desktops' verwendet werden. Das Programm funktioniert sehr gut, wenn nicht zuviele Auto-Ordner-Hintergrund-Anwendungen gestartet werden. TOS-Programme und Signum laufen problemlos.

Für Multi ST gilt im Prinzip das gleiche. Gegen Auto-Ordner ist es meinen Versuchen nach weniger empfindlich, dafür macht es aber mit einigen TOS-Programmen Schwierigkeiten. Es ist auch etwas schwieriger zu installieren und braucht gegebenenfalls mehr Speicher, da es für jeden 'Desktop' neu bootet und somit alle Programme eines Auto-Ordners erneut lädt.

*CS*

ENDE

# Extended VT52-Emulator

## Teil 4

**Zum letzten Mal möchte ich heute Ihre geschätzte Aufmerksamkeit auf den erweiterten Emulator lenken. Wie in der dritten Folge erwähnt, steht ja noch eine ESCape-Sequenz aus, mit deren Hilfe Grafiken ohne den lästigen Umweg über das VDI ausgegeben werden können. Zwei kleinere Änderungen, die die Kompatibilität des Emulators erweitern, sowie ein paar allgemeine Anmerkungen zum Thema Geschwindigkeitssteigerung von Programmen komplettieren diese Folge und beschließen damit gleichzeitig die Serie.**

### Gut geDOODLEt - Grafik per ESCape-Sequenz

Der xVT52-Emu kann Bilder im unkomprimierten Doodle-Format anzeigen, in dem jeder Bildpunkt von genau einem Bit repräsentiert wird. Die Breite des Bildes ist beliebig, muß aber eine ganze Anzahl von Spalten betragen. Das bedeutet folglich, daß die Breite - gemessen in Pixels - ein Vielfaches von acht betragen muß. Bezüglich der Höhe gibt es dagegen keine Beschränkungen; sie kann eine beliebige Anzahl von Pixels betragen. Gezeichnet wird immer nur der sichtbare Teil, d.h. über den Bildschirm hinausreichende Teile der Grafik werden unterdrückt. Die auslösende Sequenz hat folgendes Format:

*ESC r ncol nrowh nrowl data*

Dabei bedeutet ncol die Anzahl der von der Grafik benötigten Spalten plus 32 (sie erinnern sich noch an den Abschnitt über Terminal-Emulatoren?). Die beiden folgenden Bytes geben Aufschluß über die Höhe der Grafik: nrowh ist die Anzahl der vollen Hunderter plus 32, nrowl enthält den um den üblichen Offset von 32 erhöhten BCD-Wert der Zehner und Einer; data schließlich sind die Grafikdaten persönlich. Das klingt viel komplizierter als es ist, darum zum besseren Verständnis ein

### Beispiel

Nehmen wir an, Sie hätten ein Programm geschrieben und möchten als Begrüßungs-Bildschirm eine Grafik anzeigen. Diese Grafik können Sie nun mit einem beliebigen Malprogramm erzeugen und von diesem aus im Doodle-Format abspeichern (sollte das Programm dieses Format nicht kennen, so gibt es bereits eine Reihe von Accessories, mit deren Hilfe man den Bildschirminhalt als "Hardcopy" in beliebigen Formaten auf Diskette oder Platte abspeichern kann). Der Bildschirm - und damit auch die Hardcopy desselben - hat eine Auflösung von 640 mal 400 Pixels. Mit Hilfe eines kleinen Progrämmchens (z.B. dem in Listing 2) sorgt man nun einfach dafür, daß am Anfang der Hardcopy-Datei folgende Bytes eingefügt werden:

```
chr$(27)"r"chr$(80+32)
chr$(4+32)chr$(0+32).
```

Das Byte 80+32 bedeutet: Die Grafik ist 80 Spalten breit (also 640 Pixels = volle Bildschirmbreite); 4+32 steht für 400 Pixels Höhe und 0+32 bedeutet, es gibt bei der Höhe weder Zehner noch Einer. Hätte die Grafik eine Höhe von z.B. 457 Pixels, so müßte die 57 folgendermaßen kodiert werden: chr$(&H57+32). Beachten Sie bitte die hexadezimale Darstellung (&H57), weil die Zehner und Einer wie erwähnt als BCD-Ziffern übergeben werden müssen (BCD = Binary Coded Decimal). Wenn Sie die solchermaßen erweiterte Hardcopy-Datei vom Desktop anzeigen lassen, ergibt dies einen typischen Aha!-Effekt. Da die Grafik immer an der momentanen Cursor-Position ausgegeben wird, sollten Sie vor der Ausgabe sicherstellen, daß sich der Cursor auch wirklich da befindet, wo die linke obere Ecke der Grafik liegen soll (z.B. mittels ESC Y oder ESC H). Die Position des Cursors bleibt bei Grafikausgaben erhalten; ist die inverse Darstellung (ESC p) aktiv, wird die Grafik genau wie "normaler" Text invertiert!

### Grafik und DFÜ

Der im ST implementierte VT52-Emulator stellt ein Terminal-Programm dar und ist somit für Datenfernübertragung (DFÜ) geeignet. Wenn Sie also über geeignete Software verfügen, die Bildschirmausgaben nicht nur auf den Bildschirm, sondern auch auf die V.24 (bzw. RS232C) ausgibt, dann können Sie natürlich alle Ausgaben auch per Modem oder Standleitung ("Null-Modem") an einen anderen Rechner übertragen. Um z.B. zu verhindern, daß ein Rechner Daten sendet, die der andere wegen Systemüberlastung gar nicht verarbeiten kann, benötigt man ein Handshake-Protokoll. Über dieses wird angezeigt, ob ein Rechner ankommende Daten verarbeiten kann, ob er senden möchte etc.

Zur Protokoll-Realisierung gibt es zwei Möglichkeiten: Entweder man erledigt die Sache per Hardware, und zwar über zwei Leitungen, die bei der V.24 RTS und CTS genannt werden, oder aber per Software, indem man die Pegel der Lei-

tungen über zwei spezielle Signale (Bitkombinationen) XON und XOFF simuliert. Diesen beiden Signalen sind die ASCII-Codes 0x13 bzw. 0x17 zugeordnet. Wenn Sie also Daten über die serielle Schnittstelle transportieren möchten, erledigt die Treiber-Software das "Handshaking", sprich das Protokoll, automatisch entweder durch Setzen der entsprechenden Pegel an den RTS-/CTS-Leitungen oder durch Übertragen der XON/XOFF-Bytes. Gewöhnlich darf man nur einfache (ASCII-) Texte per DFÜ übermitteln, weil die ASCII-Daten unterhalb 0x20 eine besondere Bedeutung haben. Wenn Sie mit dem xVT52 Grafik-Daten übertragen möchten, müssen Sie bei aktivem XON/XOFF-Protokoll dafür sorgen, daß alle 0x13/0x17-Bytes durch (z.B.) 0x15 bzw. 0x1e ersetzt werden, weil ansonsten die Übertragung abgebrochen würde. Dies gilt natürlich weder für den Fall eines Hardware-Protokolls über RTS/CTS noch für direkte Bildschirmausgaben.
Die im xVT52-Emulator für die Grafik-Ausgabe verantwortlichen Routinen finden Sie am Ende des Listings des dritten Teils dieser Serie unter den Namen GRAPHMODE, GET_WIDTH, GET_HEIGHT_H, GET_HEIGHT_L und GRAPHICS, wobei die letztgenannte Routine die eigentliche Grafik-Ausgabe, die anderen die Vorbereitungen hierzu übernehmen.

## Trouble-Shooting

Beim Testen des neuen Emulators zeigten sich in Verbindung mit bestimmten Programmen etliche Unstimmigkeiten, die sich zumeist in zwei- bis vierbömbigen System-Verabschiedungen äußerten. Eine daraufhin vorgenommene Analyse zeigte, daß ein bestimmter Compiler Code generierte, bei dem das Datenregister D0 nach einem Aufruf des GEMDOS zur Ausgabe einer Zeichenkette ("Cconws") direkt weiterverwendet wird. Dies bedeutet, daß nicht zuerst auf evtl. aufgetretene Fehler geprüft, sondern einfach unterstellt wird, in D0 befände sich nach dem Aufruf eine Null als Langwort... Als Konsequenz wurde in der Routine UPDATE_END (1. Teil des Listings in Heft 4/88, S.129) zwischen den Zeilen 465 und 466 der Befehl "clr.l d0" eingefügt; damit liefern jetzt alle xVT52-Funktionen (außer ESC g/h) eine lange Null zurück und die von o.g. Compiler produzierten Programme sind wieder glücklich und zufrieden.

## Patch as patch can...

Wenn man einmal damit anfängt, Betriebssystem-Funktionen zu patchen, (zumal in einer so tiefgehenden Weise), sind nachträgliche Zusatz-Patches schon so gut wie vorprogrammiert.
Da der GEMDOS-Trap-Handler für die Textausgabe-Funktionen neugeschrieben wurde, klappt die Geschichte mit der Ein-/Ausgabeumlenkung "natürlich" nicht mehr. Das bedeutet z.B., daß bei Commandline-Interpretern (CLI) der Befehl "dir >prn:" die Dateien des aktuellen Verzeichnisses auf den Bildschirm statt auf den Drucker ausgibt, weil xVT52 nicht auf eventuelle Dateiumlenkungen prüfen kann. Um diesem Mißstand abzuhelfen, wurde noch eine weitere Funktion des GEMDOS abgefangen: FORCE(). Programme, die Ein-und Ausgabekanäle umlenken, bedienen sich hierzu dieser Funktion. Im Listing 3 sehen Sie den geänderten GEMDOS-Trap-Handler, den Sie bitte anstelle des im ersten Teil dieser Serie abgedruckten einsetzen (S. 127, Zeilen 90-143).
Da laut Murphy jede Fehlerbeseitigung zwei neue Fehler impliziert, hat es mich nicht sonderlich gewundert, daß (fast) alle von mir getesteten Programme auch mit dieser Version verträglich waren - bis auf die, die mit dem o.g. Compiler geschrieben wurden. Der Grund diesmal: Adreßregister A1. Obwohl nie offiziell bestätigt wurde, daß bei GEMDOS-Aufrufen nur die Register D0/A0 zerstört werden, erzeugt der Compiler einen Code, der es als größte Selbstverständlichkeit betrachtet, daß außer diesen beiden Registern alle sonstigen Werte nach einem Trap #1 erhalten bleiben... GRRRrrrrhhh!!! Nun denn, die *eine* hierdurch notwendige Änderung (das Retten von A1) ist bereits im geänderten Trap-Handler enthalten; die andere müssen Sie noch vornehmen, und zwar im ersten Teil des Listings (ST 4/88, S. 127): Ändern Sie bitte die Zeilen 73 und 80 ab in "movem.l d1-d7/a1-a6,-(a1)", bzw. "movem.l (a1)+,d1-d7/a1-a6". Wenn Sie jetzt noch Zeile 86 ("ds.w 23,0") abändern in "ds.w 31,0", ist die Welt wieder in Ordnung (es werden jetzt wirklich alle Register außer D0/A0 gerettet). Man darf gespannt sein, ob die nächste TOS-Version, die ATARI ja schon für den Herbst angedroht hat, dieselben Register bei GEMDOS-Aufrufen rettet wie in den derzeitigen Versionen. Wenn nicht, läuft kein einziges Programm mehr, das durch diesen ominösen Compiler "durchgenudelt" wurde... (wenn Sie jetzt wissen möchten, um welchen Übersetzer es hier geht, muß ich Sie enttäuschen: ich haue nur sehr ungern jemanden in die Pfanne. Es steht Ihnen natürlich frei, die Änderungen nicht vorzunehmen und erst mal zu prüfen, welche Programme unter xVT52 nicht mehr laufen).

## /AUTO/xVT52.PRG

Wo wir gerade bei Änderungen sind, bietet sich noch eine kleine Sequenz an, die nur etwa 30 Bytes zusätzlich belegt, etwas kompliziert ist und dem Emulator eine recht nützliche Eigenschaft verleiht: sie macht ihn nämlich AUTO-Ordner- und damit bootfähig.
Wenn Sie das Programm ohne die folgenden Änderungen in den AUTO-Ordner des Boot-Laufwerks legen würden, könnten Sie zwar den Emulator verwenden, müßten gleichzeitig aber auf den Cursor verzichten. Der Grund liegt hier wieder einmal in der leidigen Tatsache , daß Programme in diesem speziellen Ordner eher zur Ausführung gebracht werden als das Desktop und damit vor Installieren des GEM. Der Trap-#2-Vektor, der für VDI- und AES-Aufrufe zuständig ist, zeigt bis zur endgültigen GEM-Installation auf einen RTE-Befehl. Da xVT52 diesen Vektor rettet, würde er später alle GEM-Aufrufe ins Nirwana schicken...
Um dieses Problem zu umgehen, wird der GEM-Vektor jetzt nicht mehr direkt beim Installieren von xVT52 umgebogen, sondern erst nach der GEM-Installation. Um den richtigen Zeitpunkt zu erwischen, muß die Cursor-Interrupt-Routine (CRS_IRR) bei jedem Durchlauf prüfen, ob der Vektor immer noch auf die gleiche Adresse zeigt wie zum Zeitpunkt der Aktivierung von xVT52. Wenn ja, ist das GEM noch nicht initialisiert, andernfalls wird der nun aktive Vektor gerettet und durch den xVT52-eigenen ersetzt. Mit den folgenden paar Änderungen wird diese Vorgehensweise realisiert; sie beziehen sich alle auf den ersten Teil des Listings, wie er in der April-Ausgabe der ST-Computer (S. 126-128) zu finden ist.

1. INSTALL-Routine:
   - zwischen den Zeilen 31 und 32 einfügen: addq.l #4,a0
   - Zeile 33 ("subq.w #1,d0") ändern in: subq.w #2,d0
   - Zeilen 49 ("lea AES_VDI,a2") und 50 ("move.l a2,$88") entfernen

Die beiden ersten Änderungen sorgen dafür, daß der erste VBI-Slot über-

sprungen wird, da dieser für GEM reserviert ist; die dritte Änderung verhindert die frühzeitige Installation des eigenen Vektors.

2. CRS_IRR: - zwischen Zeilen 309 und 310 Listing 4 einfügen

   Mit diesen paar Befehlen wird sichergestellt, daß der eigene Vektor erst nach Installieren des GEM eingesetzt wird.

Das war's denn auch schon fast; zwei Bemerkungen muß ich noch los werden, nämlich *daß der so modifizierte Emulator nur noch, d.h. ausschließlich, will sagen exklusiv, beim Booten aus dem AUTO-Ordner gestartet werden darf.* Im "hochgefahrenen" Zustand führt der Versuch, xVT52 nachträglich zu starten, garantiert zu saftigen Abstürzen. Außerdem sollte der Emulator als letztes Programm im AUTO-Ordner stehen; speziell in Kombination mit weiteren System-Patches - insbesondere des neuerdings von ATARI vertriebenen TURBODOS-Programmes - kann es sonst leicht passieren, daß der Boot-Erfolg ausbleibt und - bei Verwendung einer Hard-Disk - selbige "von Hand zu Fuß" nachträglich per AHDI o.ä. hochgezogen werden muß...

## Wenn zwei das gleiche tun...

...ist das noch lange nicht dasselbe. Diese Binsenweisheit gilt auch für den erweiterten Emulator. Mit den eben vorgestellten Änderungen kann man aber von einer sehr hohen Kompatibilität zum eingebauten VT52-Emulator ausgehen. Sogar die Font-Umschaltung (Text 8/16) in GFA-Basic bereitet keinerlei Schwierigkeiten; überhaupt wird in diesem Interpreter von allen Möglichkeiten zur Text-Ausgabe Gebrauch gemacht (GEMDOS, VDI und BIOS), so daß man sich hier sehr leicht von der Kompatibilität überzeugen kann. Beim Anklicken eines Zeichens (im BASIC-Editor) mit der Maus wird dieses u.U. "unregelmäßig" invertiert; der Grund liegt darin, daß beim xVT52 die Cursor-Routine erst nach der Maus-Routine des GEM aktiviert wird, während es normalerweise umgekehrt ist. Um dieses Problemchen zu beheben, müßte man die VBL-Interrupt-Routine neuschreiben und die Cursor-Routine an der entsprechenden Stelle einklinken. Wer will, kann's ja mal versuchen (ist nicht besonders schwierig, bloß etwas aufwendig).

Ansonsten kann es nur hin und wieder vorkommen, daß der Cursor nach dem Verlassen von TOS-Anwendungen bei der Rückkehr ins Desktop stehenbleibt. Der Grund hierfür dürfte irgendwo in den Tiefen des VDI verborgen liegen, weil nicht ausgeschlossen werden kann, daß Teile des VT52-Emulators (vor allem dessen Kontroll-Variablen) vom VDI direkt angesprochen werden. Wenn es jemandem gelingen sollte, die genaue Ursache zu eruieren, wäre ich über eine entsprechende Mitteilung sehr dankbar. Überhaupt würde ich mich freuen, wenn mir die Atarianer im Lande ihre Erfahrungen, Vorschläge etc. bezüglich des erweiterten VT52-Emus zukommen ließen.

## Rev it up...

Wie versprochen, möchte ich Ihnen jetzt noch ein paar Tips verraten, mit denen Sie zeitkritische Programme auf Trab bringen. Es ist natürlich eine ziemlich triviale Feststellung, daß die Geschwindigkeit eines Programmes entscheidend vom verwendeten Algorithmus abhängt. Wenn Sie 50000 Daten per Bubblesort sortieren, können Sie alle nur erdenklichen Tricks anwenden und liegen trotzdem noch um Größenordnungen über der Laufzeit eines Quicksorts. Ebenso trivial ist die Aussage, daß ein in Assembler geschriebenes Programm nicht zwingenderweise viel schneller sein muß als ein in einer höheren Sprache geschriebenes.

Schnelle Programme lassen sich gezielt nur in Assembler realisieren, weil man in Interpreter- und Compilersprachen so gut wie keinen Einfluß auf den tatsächlich erzeugten Code hat. Die Sprache C führt hier ein Zwitterdasein, weil sie einerseits zwar typische Merkmale einer Hochsprache aufweist, andererseits aber nur einen Makro-Assembler darstellt. Gerade hier gilt: auch wenn man sich alle Mühe gibt, möglichst viele ineinandergeschachtelte Befehle zu einem einzigen zusammenzufassen: ob das noch jemand lesen kann, ist die eine Frage; ob der Compiler diese Komprimierung auch in effizienten Code umsetzen kann, bleibt dahingestellt. Aus diesen Gründen möchte ich Ihnen hier nur solche "Kniffe" näherbringen, die auch wirklich die Laufzeit optimieren, ohne das Programm in Spaghetti-Code zu verwandeln. Allerdings sollten Sie über einige C- und/oder Assembler-Kenntnisse verfügen, da die Möglichkeiten, die ich Ihnen vorstellen möchte, in erster Linie Besonderheiten des im ST verwendeten Prozessors ausnutzen.

## MOTOROLA - Moto-Roller

Und in den STs wird nun mal der MC68000 eingesetzt, der (allen Aprilscherzen zum Trotz) hier mit acht Megahertz getaktet wird. In dieser Version packt er immerhin so Daumen mal Pi etwa 300000 Befehle pro Sekunde. Geschickt programmiert, kann man damit locker etwa dreieinhalb Megabyte Speicher pro Sekunde adressieren.

Was aber ist geschickte Programmierung? Hierzu ein paar Beispiele. Wie in allen Rechnern der sogenannten "Von-Neumann-Architektur" stehen Daten wie auch Programme gleichberechtigt im Hauptspeicher. Das bedeutet, daß der Prozessor sich seine Befehle Stück für Stück aus dem Speicher holen muß, wozu er logischerweise genausolange benötigt, als ob er irgendein Datenwort lesen würde. Die Befehle, die der MC68000 versteht, sind immer zwischen zwei und zehn Bytes lang; es versteht sich von selbst, daß der Prozessor pro Zeiteinheit wesentlich mehr zwei Byte lange Befehle verarbeiten kann als solche der Länge zehn. Wenn Sie xVT52 einmal disassemblieren, können Sie erkennen, daß an allen zeitkritischen Bereichen vorzugsweise die kurzen Befehle zum Einsatz kommen. Bei Verschiebe- und Löschroutinen findet man dagegen viele MOVEM.L-Befehle. Diese benötigen zwar sechs Bytes, manipulieren hier aber (es lebe die Mikro- und Nanoprogrammierung) bis zu vierzig Bytes auf einmal - viel schneller geht's nimmer!

## Alle Register ziehen

Ein weiterer prozessorspezifischer Punkt ist die Verwendung möglichst vieler Register. Variablen, die innerhalb einer Routine oft benötigt werden, sollten immer in Registern gehalten werden. Warum? Der Zugriff auf den Hauptspeicher (RAM) ist in jedem Fall wesentlich langsamer als der auf die prozessorinternen Register. In höheren Programmiersprachen hat man in der Regel keine Möglichkeit, den Compiler zu "zwingen", bestimmte Variablen in Registern abzulegen; eine Ausnahme bilden C-Compiler, bei denen man Variablen durch Voranstellen des Schlüsselwortes "register" vor die Deklaration in Register legen lassen kann. Allerdings: Ob der

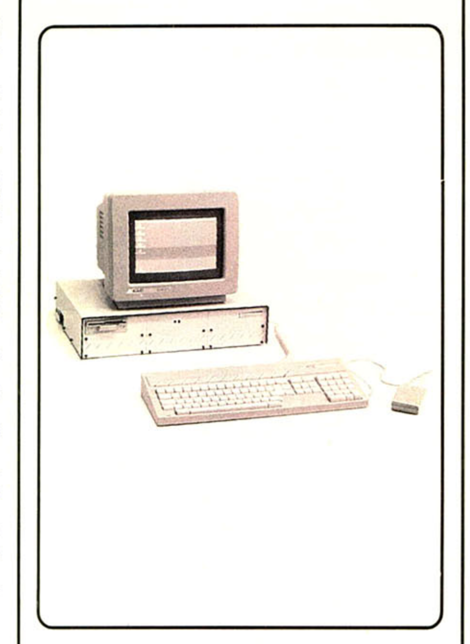
# ATARI ST ALS PREISWERTES PROFISYSTEM

### KOMPLETT-SYSTEME

Die berühmten Computer der ATARI ST-Serie, die modernste Technologien einsetzen, zu niedrigsten Preisen, sind jetzt als PROFESSIONELLES KOMPLETT-SYSTEM zu erhalten.
Das Herz dieses Systems, wie rechts abgebildet, ist der ATARI 1040 STF mit 1-MB-ARBEITSSPEICHER und einer modernen 3,5" DOPPELSEITIGER DISKDRIVE. Ein HOCHAUFLÖSENDER S/W MONITOR (SM 124), MAUS und BASIC machen die Grundversion komplett. Natürlich wird auch eine Version mit 20-MB-FESTPLATTE (SH 205) angeboten.
Die Vielfalt der PROFESSIONELLEN SOFTWARE, die nun für die ATARI-Rechner zur Verfügung steht, ist natürlich 100% lauffähig, und das flexible und geräumige Gehäuse erlaubt den Einbau von System-Erweiterungen.
Für den Kenner sind viele serienmäßige Extras eingebaut.

- Der Rechner wird beim Einschaltvorgang automatisch nach der Festplatte geordnet.
- Das sondergefertigte Hauptgehäuse verfügt über alle originalen Schnittstellen.
- Das Komplett-Gerät wird über einen Schalter geschaltet, die Festplatte kann jedoch bei Bedarf ausbleiben.
- Einbaumöglichkeit für ein weiteres 3,5"- oder 5,25"-Laufwerk und Harddisk bis zu 120 MB.
- FREIBEWEGLICHE TASTATUR mit Resetknopf und vieles mehr . . .

**L. H. 100 (System ohne Harddisk) 1998,–**
**L. H. 120 (mit 20-MB-Atari-Harddisk) 3300,–**



### BAUSÄTZE

beinhalten HAUPTGEHÄUSE, TASTATURGEHÄUSE und allen benötigten Platinen, Kabel und Kleinteile, um Ihren vorhandenen ATARI 260/520 oder 1040-Rechner, in das links beschriebene Profi-System umzubauen. Eine Umbauanleitung und technische Unterlagen sind beigefügt.
Der KOMPAKT-KIT 2 ist eine NEUENTWICKLUNG und ersetzt seinen populären Vorgänger, den die Atari-Fans seit 1½ Jahren erfolgreich einsetzen. Zu dem Umbau sind KEINERLEI LÖTARBEITEN erforderlich und er ist auch mit geringsten technischen Kenntnissen schnell und problemlos durchzuführen.
Alle HARDDISKS der Firmen ATARI und VORTEX sind ohne Zusatzteile einzubauen und unsere ZEITVERZÖGERUNG ist in jedem Kit serienmäßig dabei. Dazu passen alle gängigen 3,5"-LAUFWERKE, es wird sogar eine Blende für ein 5,25"-Laufwerk beigefügt. Bei dem KK2 260/520 wird ein SCHALTNETZTEIL mitgeliefert, dieses ersetzt das vorherige Gewirr von Netzteilen für Rechner und Floppies und erlaubt die Versorgung des kompletten Systems (außer Monitor) über ein zentrales Netzkabel und einen Schalter.
Unser FLACHES ABGESETZTES TASTATURGEHÄUSE, mit RESETKNOPF, voll entstörter Schnittstellenplatine und SPIRALKABEL, ist auch einzeln zu erhalten.

| | |
|---|---|
| **KK2 260/520** | **498,–** |
| **KK2 1040** | **398,–** |
| **Tastaturgehäuse für 260/520** | **128,–** |
| **Tastaturgehäuse für 1040** | **128,–** |

***Bestellen Sie sofort oder fordern Sie Informationen an · Bei unserem VERSAND oder bei jedem guten FACHHANDEL · OEM-Anfragen erwünscht.***




A & G SEXTON GMBH

***EINFÜHRUNGS-ANGEBOT BEI DIREKT-BESTELLUNG:***

Komplettsystem LH 120 beinhaltet:

- Atari 1040 STF, Maus u. Basic mit
- doppelseitigem Diskettenlaufwerk 3½"
- Kompaktkitgehäusesystem
- Atari SM 124 Monitor
- Atari SH 205 Harddisk

**2998 DM**

**Versand-Anschrift**
Riedstraße 2
7100 Heilbronn
Telefon 07131/78480
Telefax 07131/79778

---

Software
**T.i.M**
Time is Money

Die überarbeitete Version 1.1 unserer erfolgreichen Buchführung TiM bietet noch mehr Leistung und Bedienungskomfort. TiM, das Kernstück unserer "Time is Money" Serie, ist eine einfach zu bedienende Buchführung für den Laien und den Experten.

**TiM 1.1**
**Eine Buchführung**




Detaillierte Informationen bei Ihrem Fachhändler oder direkt bei uns.

Unverbindliche Preisempfehlung:
TiM 1.1 DM 298,–;
Handbuch DM 30,–*;
Demo 10,–;
*wird bei Direktkauf angerechnet.

**C.A.$.H. GmbH**
Robert-Bosch-Str. 20 a, 8900 Augsburg
Tel. 0821/703856

---

**Hardware**
**Software**
**Zubehör**

## Möchten Sie Ihre Umsätze steigern?

**TOP-Verkäufer-Telefon,** vers. Akquisiteur, über 18 Jahre im VK selbständig, ex. EDV-Berater, bietet Ihnen Verkaufsunterstützung in:

**Direktverkauf – Kundenberatung**

***Der Akquisiteur***
***Franz-J. Lauer***
*Eichenlaubstraße 22*
*6648 Hadern 3*
***Telefon (06871) 4515***



---

Ecosoft Economy Software AG

Postfach 30, 7701 Büsingen, Tel. 07751 - 7920

## Software (fast) gratis

- **Größtes Angebot von Prüf-Software und Frei-Programmen Europas**: Über 3'000 Disketten für IBM-PC/ Kompatible, Macintosh, Amiga, Atari ST, C64/128, Apple II.
- Viele **deutsche Programme**, speziell für Firmen, Selbständige, Privatpersonen, Schulen.
- **Software gratis**, Sie bezahlen nur eine Vermittlungsgebühr von DM 14.40 oder weniger je Diskette.
- HOTLINE: Tel. Anwenderunterstützung (3 Techniker von 8-18 Uhr).

## Verzeichnis gratis

Bitte Computermodell angeben. Gegen Einsendung dieses Inserates erhalten Sie zusätzlich einen Gutschein für eine Gratis-Diskette. 448

Compiler auch tatsächlich auf diesen frommen Wunsch eingeht oder er es dabei bewenden läßt, entscheidet er höchstselbst und nicht etwa der Programmierer...

## Relativitätstheorie

Da es sich Programme in der Regel nicht aussuchen können, an welche Speicheradresse sie zur Laufzeit vom Betriebssystem geladen werden, müssen sie praktisch an jeder (geraden) Adresse lauffähig sein. Um dies zu erreichen, gibt es beim ST zwei Möglichkeiten, von denen die eine schnell ist, dafür aber Restriktionen an die Programmlänge stellt, während der anderen Methode zwar die Länge egal ist, sie dafür aber langsamer arbeitet.
Wie die meisten Prozessoren kennt auch der MC68000 die Adressierungsart "absolut". Damit ist er in der Lage, auf jede Speicherzelle innerhalb des adressierbaren Bereiches zuzugreifen, indem er ihre "absolute" Adresse verwendet. Diese jedoch ist vier Bytes lang und kostet somit eine Menge Speicher und folglich auch Ladezeit. Für einen Befehl, der den Inhalt einer Speicherzelle in eine andere verschiebt und dabei absolut adressiert, werden volle zehn Bytes benötigt. Diese Methode bläht also Programme auf, verlangsamt sie und - was am wichtigsten ist - läßt sie generell nur an einer Adresse laufen. Denn wenn man auf eine Adresse 4711 zugreift, ist das unabhängig davon, wo das Programm steht!
Der Trick, mit dem man solchermaßen adressierende Programme trotzdem überall zum Laufen bringen kann, funktioniert folgendermaßen: Das Programm wird so kompiliert, daß es nur an Adresse Null läuft (witzig, nicht?). Im Anschluß an das eigentliche Programm steht eine Tabelle, in der alle "absoluten" Adressen aufgelistet sind; außerdem wird im Programm-Header das sogenannte Relozier-Flag gesetzt. Wird das Programm nun in den Rechner geladen und ist das Flag gesetzt, so wird auf alle gekennzeichneten Adressen die Adresse aufaddiert, an die das Programm tatsächlich geladen wurde. Verwirrt? Beispiel. Ein Programm möchte auf eine Variable zugreifen, die 5000 Bytes vom Programmanfang entfernt ist. Da der Programmstart definitionsgemäß auf Adresse 0 liegt, erhält unsere Variable die absolute Adresse 5000. Die Stelle im Programm, an der ein Befehl (z.B. MOVE) auf diese Adresse zugreift, wird in der Relo-Tabelle vermerkt. Der Befehl selbst ist jedoch für die absolute Adressierung ausgelegt und lautet z.B. MOVE.B 5000,D0 (bringe Inhalt der Speicherzelle 5000 ins Datenregister 0). Wird das Programm geladen (z.B. an Adresse 4712), addiert ein spezielles Programm des TOS (der Relocator) auf die Adresse des Befehls den Lade-Offset von 4712, so daß er nunmehr lautet: MOVE.B 9712,D0. Jetzt klarer? Diese Programme benötigen beim Laden mehr Zeit als solche der anderen Kategorie. Das ist zu verschmerzen. Sie sind aber auch länger als andere. Das wäre auch noch genehm. Sie sind langsamer als die anderen - jetzt reicht's!
Klar geht es auch anders. Anstatt nämlich auf Kosten von Speicherplatz und Performance "relativ absolut" zu adressieren, kommt man durch "pure Relativität" schneller und einfacher ans Ziel. Der "Trick" diesmal klappt wie folgt. Man sagt nicht mehr: "Adressiere ABSOLUT die Speicherzelle, die x Bytes RELATIV vom Programmstart liegt", sondern "Adressiere RELATIV die Speicherzelle, die x Bytes ABSOLUT vom momentanen Programmzählerstand entfernt ist"! Für den MOVE-Befehl liest sich das dann so, daß er den Inhalt der Speicherzelle holen soll, die x Bytes vor oder hinter der Adresse des MOVE-Befehls selbst liegt. Diese Methode der "relativen Adressierung" kennt der MC68k nämlich auch. Sie hat zwei Vor- und zwei Nachteile. Zum einen benötigt sie weniger Speicherplatz für die Befehle und arbeitet wesentlich schneller als die absolute. Zum anderen sind PC-relative Adressen bei vielen Befehlen nur als Quelloperanden erlaubt (man kann also nur den Inhalt lesen, nicht aber irgendwas reinschreiben), und der Offset, der zum aktuellen PC-Stand addiert wird, beträgt 16 Bit mit Vorzeichen. Damit kann man nur Adressen erreichen, die 32 KByte vor oder hinter dem aufrufenden Befehl liegen (verstehen Sie jetzt, warum einige Compiler Restriktionen an die Länge einer Prozedur, einer Variablendeklaration, eines Moduls etc. stellen?).
Da xVT52 aber weniger als fünf KB "verbrät", stellt dies faktisch kein Problem dar. Damit allein konnte die Geschwindigkeit schon merklich gesteigert werden. Allerdings: Diesen Kniff können Sie nur in Assembler voll ausnutzen; Compiler (Interpreter sowieso) sind nur sehr schwer davon zu überzeugen, daß man gerne effizienten Code hätte. Da xVT52 völlig PC-relativ geschrieben wurde, wollte ich diese Möglichkeit trotzdem erwähnen. Man erkennt die PC-relative Adressierung bei konventionellen 68000-Assemblern am nachgestellten (PC)-Symbol (z.B. MOVE.L DATUM(PC),D0).

## Weniger ist mehr

Einen Punkt möchte ich noch erwähnen, der die Ausführung von Schleifen beschleunigt. Hierzu eine kleine praktische Vorbetrachtung: Wenn Sie zu Hause Ihr Plattenregal säubern möchten (kann ja mal vorkommen), greifen Sie höchstwahrscheinlich nicht eine Platte nach der anderen aus dem Regal, um sie "zwischenzulagern". Anschaulich ist es klar, daß man schneller mit der Arbeit fertig ist, wenn man jedesmal ein paar Platten auf einmal anpackt und auf die Seite legt. Genauso ist es bei der Programmierung von Wiederholungsanweisungen (Schleifen). Wenn man eine Anweisung zehnmal ausführen möchte, kann man dafür zwar schreiben "for i=1 to 10 anw", dann muß aber außer der eigentlichen Anweisung der Schleifenkonstrukt "mitgeschleppt" werden: nach jedem Durchlauf muß die Laufvariable inkrementiert und auf die Ende-Bedingung überprüft werden. Daß so etwas u.U. sehr zeitintensiv ist, versteht sich von selbst. Aus diesem Grund sollte man besser die Anzahl der Schleifendurchläufe verringern und dafür die auszuführende Anweisung ein paarmal hintereinander schreiben (also etwa "for i=1 to 2 {anw; anw; anw; anw; anw}"). Bei xVT52 finden in allen zeitkritischen Routinen (etwa zum Darstellen/Invertieren von Zeichen) nur die "verkürzten" Schleifen Verwendung.
Das sind zwar bei weitem nicht alle Möglichkeiten, die zur Beschleunigung von Programmen zur Verfügung stehen, aber es sind zumindest sehr effiziente Methoden, die auch im erweiterten Emulator eingesetzt wurden. Und daß dieser recht flott arbeitet, belegen die Benchmarks der letzten Folge - gell?

## Bye bye

So, nachdem ich Sie nun vier Monate mit meinem Programm bei Laune gehalten habe, wird es höchste Zeit, mich zu verabschieden. Ich hoffe, daß Sie regen Gebrauch von den neuen Ausgabefähigkeiten machen werden und Sie im Verlauf dieser Serie das eine oder andere dazugelernt und sich vielleicht auch etwas Appetit für eigene (Assembler-)

Projekte geholt haben. Sollten Sie zum Programm oder den im Verlauf dieser Serie angesprochenen Themen Fragen und/oder Antworten haben, können Sie mir ja mal schreiben (via ST-Redaktion). Fänd' ich nett.

Zum guten Schluß habe ich Ihnen in Listing 5 noch ein kleines GFA-Progrämmchen vorbereitet, mit dessen Hilfe Sie ausprobieren können, ob Sie den Emulator auch richtig eingegeben haben; es erzeugt eine kleine Datei, die Sie sich bei installiertem xVT52 vom Desktop aus anzeigen lassen sollten...

Viel Spaß und - tschüß!

*MS*

```
  1: ; ************************************
  2: ; * >EXTENDED VT52-TERMINAL EMULATOR< *    4. Teil (Ende)
  3: ; *==================================*
  4: ; * Entwickelt mit PROFIMAT-Assembler *
  5: ; *        M.Schumacher, (p) 12/87     *
  6: ; ************************************
  7:
  8:
  9:
 10: FSCR_LEFT:                     ; ESC 'T'
 11:   move.l   6(a4),a1            ; abs. Cursorposition
 12:   suba.w   (a0),a1             ; -akt. Spalte=^Zeilenanfang
 13:   adda.w   38(a0),a1           ; +Bytes/Textzeile=^Anfang
                                   nächster Textzeile
 14:   move.w   36(a0),d2           ; Zeichenhöhe (Pixels)
 15:   subq.w   #1,d2               ; in dbra-Zähler wandeln
 16: \lp1:
 17:   moveq    #3,d1               ; 4*10 Worte verschieben
 18:   move.w   #0,CCR              ; SR-Flags auf Null
 19:   btst     #7,-80(a1)          ; Pixel ganz links gesetzt?
 20:   beq.s    \lp2                ; nein
 21:   move.w   #16,CCR             ; sonst X-Bit setzen
 22: \lp2:
 23:   roxl     -(a1)
 24:   roxl     -(a1)
 25:   roxl     -(a1)
 26:   roxl     -(a1)
 27:   roxl     -(a1)
 28:   roxl     -(a1)
 29:   roxl     -(a1)
 30:   roxl     -(a1)
 31:   roxl     -(a1)
 32:   roxl     -(a1)
 33:   dbra     d1,\lp2
 34:   dbra     d2,\lp1     ; nächste Pixelzeile verschieben
 35:   rts
 36:
 37: FSCR_RIGHT:                    ; ESC 'U'
 38:   move.l   6(a4),a1            ; abs. Cursorposition
 39:   suba.w   (a0),a1             ; -akt. Spalte=^Zeilenanfang
 40:   move.w   36(a0),d2           ; Zeichenhöhe (Pixels)
 41:   subq.w   #1,d2               ; in dbra-Zähler wandeln
 42: \lp1:
 43:   moveq    #3,d1               ; 4*10 Worte verschieben
 44:   move.w   #0,CCR              ; SR-Flags auf Null
 45:   btst     #0,79(a1)           ; Pixel ganz rechts gesetzt?
 46:   beq.s    \lp2                ; nein
 47:   move.w   #16,CCR             ; sonst X-Bit setzen
 48: \lp2:
 49:   roxr     (a1)+
 50:   roxr     (a1)+
 51:   roxr     (a1)+
 52:   roxr     (a1)+
 53:   roxr     (a1)+
 54:   roxr     (a1)+
 55:   roxr     (a1)+
 56:   roxr     (a1)+
 57:   roxr     (a1)+
 58:   roxr     (a1)+
 59:   dbra     d1,\lp2
 60:   dbra     d2,\lp1     ; nächste Pixelzeile verschieben
 61:   rts
 62:
 63: FDSCRUP:                       ; ESC 'W'
 64:   move.w   6(a0),d1            ; max. Zeile
 65:   sub.w    2(a0),d1            ; - aktuelle
 66:   move.l   6(a4),a1            ; abs. Cursorposition
 67:   suba.w   (a0),a1             ; - akt. Spalte = ^Zeilenanfang
 68:   bra.s    FSCRUP_ENTRY        ; schieben+löschen
 69:
 70: FSCRUP:                        ; ESC 'X'; Bildschirminhalt
                                   incl. der akt. Zeile
 71:                                ; um eine Zeile pixelweise
                                   nach oben schieben
 72:   move.l   LOGBASE,a1          ; ^Video-RAM
 73:   move.w   2(a0),d1            ; aktuelle Zeile
 74: FSCRUP_ENTRY:                  ; Einsprung für ESC 'W'
 75:   move.w   d1,d0               ; Zeile retten
 76:   addq.w   #1,d0               ; plus 1 = Anzahl Textzeilen
 77:   move.w   36(a0),d1           ; Zeichenhöhe in Pixels
 78:   mulu     d1,d0               ; Anzahl Textzeilen mal
                                   Zeichenhöhe
```

```
 79:   subq.w   #1,d0               ; 1 Zeile abziehen
 80:   lsr.w    #1,d0               ; div 2 (es werden immer 2
                                   Pixelzeilen verschoben)
 81:   subq.w   #1,d0               ; in dbra-Zähler wandeln
 82:   move.w   d0,d2               ; und merken
 83:   subq.w   #1,d1               ; Zeichenhöhe in dbra-Zähler
                                   wandeln
 84:   movem.l  a0/a4/a6,-(a7)      ; Register retten
 85:   move.l   a1,-(a7)            ; Startadresse merken
 86: \lp_scr:
 87:   movea.l  (a7),a1             ; Ziel
 88:   movea.l  a1,a2
 89:   adda.w   #80,a2              ; Quelle liegt 1 Pixelzeile
                                   tiefer
 90:   move.w   d2,d0               ; Anzahl zu scrollender
                                   Pixelzeilen-1
 91: \scr_ln:                       ; Bildschirminhalt um 1 Pixel
                                   nach oben scrollen
 92:   movem.l  (a2)+,REGISTER
 93:   movem.l  REGISTER,(a1)
 94:   movem.l  (a2)+,REGISTER
 95:   movem.l  REGISTER,40(a1)  ; 2 Pixelzeilen verschieben
 96:   movem.l  (a2)+,REGISTER
 97:   movem.l  REGISTER,80(a1)
 98:   movem.l  (a2)+,REGISTER
 99:   movem.l  REGISTER,120(a1)
100:   adda.w   #160,a1
101:   dbra     d0,\scr_ln          ; die nächsten beiden
                                   Pixelzeilen verschieben
102:   movem.l  (a2)+,REGISTER      ; die letzte Pixelzeile
                                   verschieben
103:   movem.l  REGISTER,(a1)
104:   movem.l  (a2)+,REGISTER
105:   movem.l  REGISTER,40(a1)
106:   movem.w  ZEROES,REGISTER     ; Register löschen
107:   movem.l  REGISTER,80(a1)     ; und letzte Pixelzeile damit
                                   löschen
108:   movem.l  REGISTER,120(a1)
109:   dbra     d1,\lp_scr          ; Bildschirm erneut um 1
                                   Pixel nach oben scrollen
110:   addq.l   #4,a7               ; Startadresse wegwerfen
111:   movem.l  (a7)+,a0/a4/a6      ; Register zurück
112:   rts
113:
114: ZEROES:
115:   ds.l     5,0                 ; 5 Null-Langworte zum Löschen
                                   der Register
116:
117: FDSCRDN:                       ; ESC 'Z'
118:   move.w   2(a0),d1            ; akt. Zeile
119:   move.l   6(a4),a1            ; abs. Cursorposition
120:   suba.w   (a0),a1             ; ^Anfang der Textzeile
121:   adda.w   38(a0),a1           ; ^Anfang der nächsten
                                   Textzeile
122:   suba.w   #80,a1              ; ^Anfang der letzen Pixelzeile
                                   der Textzeile (Puh!)
123:   bra.s    FSCRDN_ENTRY        ; schieben+löschen
124:
125: FSCRDN:                        ; ESC 'V'; Bildschirminhalt
                                   incl. der akt. Zeile
126:                                ; um eine Zeile pixelweise
                                   nach unten schieben
127:   move.l   LOGBASE,a1          ; ^Video-RAM
128:   adda.l   #32000-80,a1        ; ^letzte Pixel-Zeile
129:   move.w   6(a0),d1            ; max. Zeile
130:   sub.w    2(a0),d1            ; -aktuelle Zeile = Anzahl zu
                                   scrollender Zeilen-1
131: FSCRDN_ENTRY:                  ; Einsprung für ESC 'Z'
132:   addq.w   #1,d1               ; +1=Anzahl zu scrollender
                                   Textzeilen
133:   move.w   d1,d0               ; Anzahl Zeilen retten
134:   move.w   36(a0),d1           ; Zeichenhöhe in Pixels
135:   mulu     d1,d0               ; mal Anzahl Textzeilen
136:   subq.w   #1,d0               ; minus 1 Pixelzeile
137:   lsr.w    #1,d0               ; div 2 (es werden immer 2
                                   Pixelzeilen verschoben)
138:   subq.w   #1,d0               ; in dbra-Zähler wandeln
139:   move.w   d0,d2               ; und merken
140:   subq.w   #1,d1               ; Zeichenhöhe in dbra-Zähler
                                   wandeln
141:   movem.l  a0/a4/a6,-(a7)      ; Register retten
```

```
142:  move.l   a1,-(a7)          ; Startadresse merken
143: \lp_scr:
144:  movea.l  (a7),a1           ; Ziel
145:  movea.l  a1,a2
146:  suba.w   #80,a2            ; Quelle liegt 1 Pixelzeile
                                   höher
147:  move.w   d2,d0             ; Anzahl zu scrollender
                                   Pixelzeilen-1
148: \scr_ln:                    ; Bildschirminhalt um 1 Pixel
                                   nach unten scrollen
149:  movem.l  (a2)+,REGISTER
150:  movem.l  REGISTER,(a1)
151:  movem.l  (a2),REGISTER
152:  movem.l  REGISTER,40(a1)   ; 2 Pixelzeilen verschieben
153:  suba.w   #120,a2
154:  movem.l  (a2)+,REGISTER
155:  movem.l  REGISTER,-80(a1)
156:  movem.l  (a2),REGISTER
157:  movem.l  REGISTER,-40(a1)
158:  suba.w   #120,a2
159:  suba.w   #160,a1
160:  dbra     d0,\scr_ln        ; die nächsten beiden
                                   Pixelzeilen verschieben
161:  movem.l  (a2)+,REGISTER    ; die letzte Pixelzeile
                                   verschieben
162:  movem.l  REGISTER,(a1)
163:  movem.l  (a2),REGISTER
164:  movem.l  REGISTER,40(a1)
165:  movem.w  ZEROES,REGISTER   ; Register löschen
166:  movem.l  REGISTER,-80(a1); und letzte Pixelzeile damit
                                   löschen
167:  movem.l  REGISTER,-40(a1)
168:  dbra     d1,\lp_scr        ; nächste Textzeile
169:  addq.l   #4,a7             ; Startadresse wegwerfen
170:  movem.l  (a7)+,a0/a4/a6    ; Register zurück
171:  rts
172:
173: DIR_CRS:                    ; ESC 'Y'
174:  lea      VEC_BASE,a1       ; Vektor auf Holen
175:  lea      ESC_Y_X,a0        ; der Zeile
176:  move.l   a0,(a1)           ; umbiegen
177:  rts                        ; fertig
178: ESC_Y_X:
179:  lea      TCB,a0            ; ^TerminalControlBlock
180:  lea      VEC_BASE,a1       ; Vektor
181:  lea      ESC_Y_Y,a2        ; auf Y-Angabe
182:  move.l   a2,(a1)           ; umschalten
183:  subi.w   #32,d0            ; Offset abziehen
184:  bmi.s    \zurück           ; zu klein, ignorieren
185:  cmp.w    6(a0),d0          ; größer als größte Zeile?
186:  bgt.s    \zurück           ; ja, ignorieren
187:  move.w   d0,2(a0)          ; sonst Zeile setzen
188: \zurück:
189:  rts                        ; fertig
190: ESC_Y_Y:
191:  lea      TCB,a0            ; ^TerminalControlBlock
192:  lea      VEC_BASE,a1       ; Vektor wieder
193:  lea      STD_VEC,a2        ; auf normale Ausgabe
194:  move.l   a2,(a1)           ; umschalten
195:  bsr      UPDATE_CRS        ; Cursor ausschalten
196:  subi.w   #32,d0            ; Offset von Spaltenwert
                                   abziehen
197:  bmi.s    \zurück           ; zu klein, ignorieren
198:  cmpi.w   #80,d0            ; größer als größte Spalte?
199:  bpl.s    \zurück           ; ja, ignorieren
200:  move.w   d0,(a0)           ; sonst Spalte setzen
201: \zurück:
202:  bsr      UPDATE_END        ; und Cursor wieder einschalten
203:  rts                        ; fertig
204:
205:
206:
207: ; **********************************
208: ; * VDI-INTERFACE FÜR xVT52-EMULATOR *
209: ; **********************************
210:
211: VDI_ENTRY:
212:  move.l   d1,-(a7)          ; ^ParameterBlock retten
213:  bsr      UPDATE_CRS        ; Cursor ausschalten
214:  move.l   (a7)+,a3          ; ^ParameterBlock
215:  lea      TCB,a0            ; ^TerminalControlBlock
216:  move.l   (a3),a2           ; ^Contrl-Array
217:  move.w   10(a2),d0         ; Funktionsnummer holen
218:  cmpi.w   #101,d0           ; VDI ESC 101?
219:  bne.s    \v102             ; nein
220:  bsr      TOP_OFFSET        ; sonst Offset zum Bildschirm
                                   -Anfang setzen
221:  bra      UPDATE_END        ; Cursor freigeben und zurück
222: \v102:
223:  cmpi.w   #102,d0           ; VDI ESC 102?
224:  bne.s    \vdi_norm         ; nein
225:  bsr      INIT_FONT         ; sonst Font initialisieren
226:  bra      UPDATE_END        ; Cursor freigeben und zurück
227: \vdi_norm:
228:  add.w    d0,d0             ; Funktionsnummer*2
                                   (Word-Pointer)
229:  lea      DUMMY,a1          ; Adresse der 1. Routine
230:  lea      VDI_ROUTS,a2      ; Adresse der Adress-Tabelle
231:  move.w   0(a2,d0.w),a2     ; rel. Adresse der Routine
232:  adda.l   a1,a2             ; +Offset=abs. Adresse der
                                   Routine
233:  jsr      (a2)              ; selbige zur Ausführung
                                   bringen
234:  bra      UPDATE_END        ; Cursor freigeben und zurück
235:
236:
237: VDI_ROUTS:                  ; NR. FUNKTION
238:  dc.w 0                     ;  0, unbelegt
239:  dc.w GET_SIZE-DUMMY        ;  1, inquire screen size
240:  dc.w EXIT_A-DUMMY          ;  2, exit alpha mode
                                   (ESC 'f'+ESC 'E')
241:  dc.w ENTER_A-DUMMY         ;  3, enter alpha mode
                                   (ESC 'E'+ESC 'e')
242:  dc.w CRS_UP-DUMMY          ;  4, cursor up    (ESC 'A')
243:  dc.w CRS_DOWN-DUMMY        ;  5, cursor down  (ESC 'B')
244:  dc.w CRS_RIGHT-DUMMY       ;  6, cursor right (ESC 'C')
245:  dc.w CRS_LEFT-DUMMY        ;  7, cursor left  (ESC 'D')
246:  dc.w HOME-DUMMY            ;  8, home         (ESC 'H')
247:  dc.w DEL_FROM_CRS-DUMMY    ;  9, erase to end of screen
                                   (ESC 'J')
248:  dc.w L_TO_END-DUMMY        ; 10, erase to end of line
                                   (ESC 'K')
249:  dc.w GOTOXY-DUMMY          ; 11, set cursor position
250:  dc.w WR_TXT-DUMMY          ; 12, write text
251:  dc.w INV_ON-DUMMY          ; 13, inverse mode on
                                   (ESC 'p')
252:  dc.w INV_OFF-DUMMY         ; 14, inverse mode off
                                   (ESC 'q')
253:  dc.w GET_CRSPOS-DUMMY      ; 15, inquire cursor position
254:
255:
256:
257: DUMMY:    ; Leer-Routine für nicht benutzte Sequenzen
258:  rts
259:
260: GET_SIZE:                   ; VDI ESC 1
261:  move.l   (a3),a5           ; ^CONTRL-Block
262:  move.w   #2,8(a5)          ; 2 Rückgabewerte in INTOUT
263:  move.l   12(a3),a5         ; ^INTOUT-Block
264:  move.w   #80,2(a5)         ; max. Spalte
265:  move.w   6(a0),(a5)        ; max. Zeile
266:  addq.w   #1,(a5)           ; +1
267:  rts
268:
269: EXIT_A:                     ; VDI ESC 2
270:  bclr     #3,(a4)           ; Cursor ausschalten
271:  bra      CLS               ; Bildschirm löschen und zurück
272:
273: ENTER_A:                    ; VDI ESC 3
274:  bsr      CLS               ; Bildschirm löschen
275:  bset     #3,(a4)           ; Cursor einschalten
276:  rts
277:
278: GOTOXY:                     ; VDI ESC 11
279:  move.l   4(a3),a5          ; ^INTIN-Array
280:  move.w   (a5),d1           ; Zeile
281:  subq.w   #1,d1             ; -Offset
282:  bmi.s    \col              ; <0, ignorieren
283:  cmp.w    6(a0),d1          ; zu groß?
284:  bgt.s    \col              ; ja, ignorieren
285:  move.w   d1,2(a0)          ; sonst übernehmen
286: \col:
287:  move.w   2(a5),d1          ; Spalte
288:  subq.w   #1,d1             ; -Offset
289:  bmi.s    \zurück           ; <0, ignorieren
290:  cmpi.w   #79,d1            ; >79?
291:  bgt.s    \zurück           ; ja, ignorieren
292:  move.w   d1,(a0)           ; Spalte übernehmen
293: \zurück:
294:  rts
295:
296: WR_TXT:                     ; VDI ESC 12
297:  move.l   (a3),a5           ; ^CONTRL-Block
298:  move.w   6(a5),d1          ; Anzahl auszugebender Zeichen
299:  subq.w   #1,d1             ; in dbra-Zähler wandeln
300:  bmi.s    \zurück           ; <0, aufhören
301:  move.l   4(a3),a5          ; ^INTIN-Block
302: \lp:
303:  move.w   (a5)+,d0          ; Zeichen holen
304:  movem.l  d1/a5,-(a7)       ; Register retten
305:  bsr      CON_ENTRY         ; Zeichen ausgeben
306:  movem.l  (a7)+,d1/a5       ; Register zurück
307:  dbra     d1,\lp            ; nächstes Zeichen ausgeben
308: \zurück:
309:  rts
```

```
310:
311:  GET_CRSPOS:                    ; VDI ESC 15
312:   move.l    (a3),a5             ; ^CONTRL-Block
313:   move.w    #2,8(a5)            ; 2 Rückgabewerte in INTOUT
314:   move.l    12(a3),a5           ; ^INTOUT-Block
315:   move.w    2(a0),(a5)          ; akt. Zeile
316:   addq.w    #1,(a5)+            ; +Offset
317:   move.w    (a0),(a5)           ; akt. Spalte
318:   addq.w    #1,(a5)             ; +Offset
319:   rts
320:
321:  TOP_OFFSET:                    ; VDI ESC 101
322:   move.l    4(a3),a5            ; ^INTIN-Block
323:   move.w    (a5),d0             ; Offset in Pixelzeilen holen
324:   mulu      38(a0),d0           ; *Bytes/Textzeile
325:   move.l    LINE_A,a2           ; ^Line A-Block
326:   move.w    d0,-$1E(a2)         ; Byte-Offset in VDI-Block
                                      eintragen
327:   rts
328:
329:  INIT_FONT:                        ; VDI ESC 102
330:   move.l    4(a3),a5               ; ^INTIN-Block
331:   move.l    LINE_A,a6              ; ^LINE A-Block
332:   move.l    (a5),a5                ; ^Font-Header (INTIN[0,1])
333:   move.w    82(a5),d0              ; Zeichenhöhe in Pixel
334:   move.w    d0,36(a0)              ; im eigenen und
335:   move.w    d0,-$2E(a6)            ; im VDI-Block merken
336:   moveq     #80,d1                 ; Bytes pro Pixelzeile
337:   mulu      d0,d1                  ; 80 * Zeichenhöhe
338:   move.w    d1,38(a0)              ; = Bytes pro Textzeile
339:   move.w    d1,-$28(a6)            ; im VDI-Block merken
340:   move.l    #400,d1                ; 400 Pixelzeilen/Bildschirm
341:   divu      d0,d1                  ; div Zeichenhöhe
342:   subq.w    #1,d1                  ; - 1
343:   move.w    d1,6(a0)               ; = größte Zeilennummer
344:   move.w    d1,-$2A(a6)            ; im VDI-Block merken
345:   move.l    #640,d1                ; 640 Pixels pro Pixelzeile
346:   divu      52(a5),d1              ; div Zeichenbreite in Pixels
347:   subq.w    #1,d1                  ; - 1
348:   move.w    d1,-$2C(a6)            ; = größte Spaltennummer
349:   move.l    72(a5),-$A(a6)         ; Zeiger auf Offset-Tabelle
350:   move.w    80(a5),-$E(a6)         ; Breite eines Zeichens
                                         (MUSS 8 Bit betragen!!!)
351:   move.w    36(a5),-$10(a6)        ; ASCII-Code des ersten
                                         darstellbaren Zeichens
352:   move.w    38(a5),-$12(a6)        ; ASCII-Code des letzten dar.
                                         Zeichens im Font
353:   move.l    76(a5),a5              ; ^Fontdaten
354:   move.l    a5,-$16(a6)            ; im VDI-Block merken
355:   move.l    a5,32(a0)              ; und zum aktuellen VT52-Font
                                         machen
356:   subq.w    #8,d0                  ; 8x8-Font?
357:   bne.s     \8x16                  ; nein
358:   move.l    a5,28(a0)              ; ^8x8-Fontdaten
359:   bra.s     \init_crs              ; Cursor initialisieren
360:   rts
361:  \8x16:
362:   subq.w    #8,d0                  ; 8x16-Font?
363:   bne.s     \zurück                ; nein, Chaos-Format (nicht
                                         darstellbar...)
364:   move.l    a5,24(a0)              ; ^8x16-Fontdaten
365:  \init_crs:
366:   clr.l     (a0)                   ; Cursor in Home-Position
                                         bringen
367:   clr.w     10(a0)                 ; gemerkte Cursorpositionen
                                         löschen
368:  \zurück:
369:   rts
370:
371:
372:
373:   data
374:
375:  REDIR:     dc.w 0                 ; Flag für Umlenkung (0=nein)
376:  AES_OK:    dc.w 0                 ; Flag, ob AES
                                         installiert (0=nein)
377:
378:  TRAPS:                            ; Adressen der Original-GEMDOS
                                         -Trap-Handler
379:            dc.l 0          ;  0(TRAPS) GEMDOS
380:            dc.l 0          ;  4(TRAPS) AES/VDI
381:            dc.l 0          ;  8(TRAPS) BIOS
382:            dc.l 0          ; 12(TRAPS) XBIOS
383:
384:  LINE_A:   dc.l 0          ; ^LINE A-Block
385:  PBLOCK:   dc.l 0          ; Speicher für
                                  VDI-Parameterblock
386:
387:  VEC_BASE: dc.l 0          ; ^Verarbeitungsroutine für
                                  Zeichen
388:  BEL_ADR:  dc.l 0          ; Speicher für Adresse der
                                  Bimmel-Routine
389:
390:  TCB:                      ; Terminal Control Block
391:            dc.w 0          ;  0(TCB) akt. Spalte
392:            dc.w 0          ;  2(TCB) akt. Zeile
393:            dc.w 79         ;  4(TCB) größte Spalte
394:            dc.w 24         ;  6(TCB) größte Zeile
395:            dc.b %0000      ;  8(TCB) Bit 0=1: Invertieren
                                                   eingeschaltet
396:                            ;         Bit 1=1: Wrapping
                                                   eingeschaltet
397:                            ;         Bit 2=1: Unterstrich
                                                   eingeschaltet
398:                            ;         Bit 3=1: Halbe
                                                   Helligkeit
                                                   eingeschaltet
399:            dc.b 0          ;  9(TCB) Flag für Grafik
                                          (0=aus, -1=ein)
400:            dc.w 0          ; 10(TCB) Anzahl gespeicherter
                                          Cursorpositionen
401:            dc.w 0,0        ;         1. gespeicherte
                                          Position (x,y)
402:            dc.w 0,0        ;         2. gespeicherte
                                          Position (x,y)
403:            dc.w 0,0        ;         3. gespeicherte
                                          Position (x,y)
404:            dc.l 0          ; 24(TCB) ^8x16 Font (GEM)
405:            dc.l 0          ; 28(TCB) ^8x8 Font (GEM)
406:            dc.l 0          ; 32(TCB) ^aktueller Font
407:            dc.w 16         ; 36(TCB) Höhe eines Zeichens
                                          in Pixels
408:            dc.w 16*80      ; 38(TCB) Bytes pro Textzeile
409:
410:                            ; *** GRAFIK-VARIABLEN: ***
411:
412:            dc.w 0          ; 40(TCB) Anzahl möglicher
                                          horizontaler Bytes
413:            dc.w 0          ; 42(TCB) Anzahl möglicher
                                          Pixelzeilen
414:            dc.w 0          ; 44(TCB) Grafikbreite
                                          (in Bytes)
415:            dc.w 0          ; 46(TCB) Grafikhöhe
                                          (in Pixels)
416:            dc.w 0          ; 48(TCB) Zähler (horizontal)
417:            dc.w 0          ; 50(TCB) Zähler (Pixelzeilen)
418:            dc.l 0          ; 52(TCB) linker Offset (abs.)
419:                                      für Pixelzeile
420:  TABS:     ds.w 5,$8080    ; Bitvektor für Tabulatoren
421:
422:  CCB:      dc.b %0010      ; Cursor Control/Status Block
423:                            ;    Bit 0=1: Cursor enable
                                              (für IRR)
424:                            ;    Bit 1=1: Cursor darf
                                              blinken
425:                            ;    Bit 2=1: Cursorposition
                                              invertiert
426:                            ;    Bit 3=1: Cursor
                                              eingeschaltet
427:            dc.b 0          ;  1(CCB) Anzahl gespeicherter
                                          CUR_OFFs (ESC 'f')
428:            dc.w 20         ;  2(CCB) Blinkrate
429:            dc.w 20         ;  4(CCB) Zähler für Blinkrate
430:            dc.l 0          ;  6(CCB) Cursorposition
                                          (absolut)
431:
432:    end
```

*Listing 1: Der vierte Teil des xVT52-Emulators (Ende)*

```
 1:  ' *** xVT52/DOODLE-Konverter ***
 2:  '
 3:  e$=CHR$(27)
 4:  a$=SPACE$(32000)
 5:  BLOAD "eindat.doo",VARPTR(a$)
 6:  a$=e$+"f"+e$+"rp$ "+a$+e$+"a"
 7:  BSAVE "ausdat.dat",VARPTR(a$),LEN(a$)
 8:  '
 9:  ' Für 'eindat.doo' bzw. 'ausdat.dat'
10:  ' sind die entsprechenden Namen der
11:  ' gewünschten Dateien zusetzten.
```

*Listing 2: Dieses Progrämmchen wandelt DOODLE-Hardcopies für xVT52 um, so daß sie direkt vom Desktop angezeigt werden können.*

```
 1:  GEMDOS:
 2:   movea.l  a7,a0        ; A0=SSP;
 3:   btst     #5,(a0)      ; if (Aufruf aus S-Mode)
 4:   beq.s    \from_user
 5:   addq.l   #6,a0        ; then Offset addieren (PC.L+SR.W)
 6:   bra.s    \test_pline
 7:  \from_user:
 8:   move.l   USP,a0       ; else User Stack benutzen;
 9:  \test_pline:
10:   movem.l  a0,-(a7)     ; A0 retten
11:   lea      REDIR,a0     ; Zeiger auf Flag für Ausgabe
                              -Umlenkung
```

```
12:   tst.w     (a0)          ; gesetzt?
13:   movem.l   (a7)+,a0      ; A0 zurück,
                                keine CCR-Beeinflussung
14:   beq.s     \test_gdos    ; keine Umlenkung => auf Ausgabe-
                                funktionen prüfen
15:   cmpi.w    #$46,(a0)     ;   FORCE-Aufruf?
16:   bne       \orig         ;   nein
17:   cmpi.w    #1,2(a0)      ;   Umlenkung auf Stdout?
18:   bne       \orig         ;   nein
19:   lea       REDIR,a0      ;   sonst Flag wieder
20:   clr.w     (a0)          ;   auf 0 setzen
21:   bra       \orig         ;   ab ins GEMDOS
22: \test_gdos:
23:   cmpi.w    #9,(a0)       ; if (Funktion == PRINT LINE)
24:   beq.s     \pline        ; then eigene Funktion benutzen
25:   cmpi.w    #64,(a0)      ; else if (Funktion == WRITE)
26:   beq.s     \write        ; then auf Ausgabe-Kanal prüfen
27:   cmpi.w    #2,(a0)       ; else if (Funktion == CCONOUT)
28:   beq.s     \cconout      ; then eigene Funktion benutzen
29:   cmpi.w    #6,(a0)       ; else if (Funktion != CRAWIO)
30:   bne.s     \test_force   ; then auf force() testen
31:   cmpi.b    #-1,3(a0)     ; else if (Zeichen == 255)
32:   beq.s     \orig         ; then Original aufrufen
33: \test_force:
34:   cmpi.w    #$46,(a0)     ; FORCE-Aufruf?
35:   bne.s     \orig         ; nein
36:   cmpi.w    #1,2(a0)      ; wird auf Stdout umgelenkt?
37:   bne.s     \orig         ; nein
38:   lea       REDIR,a0      ; sonst Umlenkungs-Flag
39:   st        (a0)          ; setzen
40:   bra.s     \orig         ; und GEMDOS aufrufen
41: \cconout:
42:   bsr       SAVE_REGS     ; Register retten
43:   bsr       CON_OUT       ; Zeichen ausgeben
44:   bra       REST_REGS     ; Register zurück, fertig
45: \pline:
46:   bsr       SAVE_REGS     ; Register retten
47:   bsr       WRITE         ; String ausgeben
48:   bra       REST_REGS     ; Register zurück, fertig
49: \orig:
50:   move.l    TRAPS,a0      ; else Originalroutine
51:   jmp       (a0)          ; benutzen
52: \write:
53:   cmpi.w    #1,2(a0)      ; Ausgabe auf CON:?
54:   bne.s     \orig         ; nein, ham wa nix mit zu tun
55:   addq.w    #2,a0         ; sonst Anzahl auszugebender
                                Zeichen
56:   move.l    2(a0),d1      ; holen
57:   beq.s     \retour       ; kein Zeichen ist definitiv zu
                                wenig!
58:   bsr       SAVE_REGS .   ; ansonsten Register retten
59:   addq.w    #4,a7         ; (a7) = String-Pointer
60:   move.l    d1,-(a7)      ; Anzahl auf Stack retten
61:   move.l    4(a7),-(a7)   ; String-Pointer auf Stack legen
62: \lp:
63:   move.l    (a7),a0       ; String-Pointer holen
64:   addq.l    #1,(a7)       ; auf nächstes Zeichen zeigen
                                lassen
65:   move.b    (a0),d1       ; Zeichen holen
66:   and.w     #$FF,d1       ; nur LSB beachten
67:   move.w    d1,-(a7)      ; Zeichen auf Stack legen
68:   bsr       CON_OUT       ; und ausgeben
69:   addq.w    #2,a7         ; Stack korrigieren
70:   subq.l    #1,4(a7)      ; Anzahl Zeichen dekrementieren
71:   bne.s     \lp           ; und nächstes Zeichen ausgeben
72:   addq.l    #8,a7         ; Zähler und Pointer löschen
73:   bra       REST_REGS     ; Register restaurieren und
                                zurückspringen
74: \retour:
75:   rte
```

*Listing 3: Neuer GEMDOS-Trap-Handler für xVT52 zum Abfangen von Ein-/Ausgabeumlenkungen*

```
 1: ;CRS_IRR:                   ; folgende Zeilen bitte einfügen
                                (s. Text)
 2:   lea      AES_OK,a1        ; ^Flag laden
 3:   tst.w    (a1)             ; AES aktiv?
 4:   bne.s    \aes_ok          ; ja, normal weitermachen
 5:   lea      TRAPS+4,a2       ; ^alter AES/VDI-Vektor
 6:   move.l   $88,a0           ; aktueller AES/VDI-Vektor
 7:   cmpa.l   (a2),a0          ; noch derselbe?
 8:   beq.s    \irr_end         ; ja, GEM noch nicht initialisiert
 9:   move.l   a0,(a2)          ; sonst Originaladresse AES/VDI
                                merken
10:   st       (a1)             ; Flag setzen (für: AES aktiv)
11:   lea      AES_VDI,a0       ; und eigene Routine
12:   move.l   a0,$88           ; installieren
13:   bra.s    \irr_end         ; fertig, ab jetzt geht's normal
                                weiter
14: \aes_ok:
15:   ; ab hier geht es wieder mit Zeile 310 ff
      ("move.l $88,d1") weiter
```

*Listing 4: Der "GEM-Schlüssel" zum AUTO-Ordner-Problem*

```
 1: ' *** xVT52-Demobild-Generator ***
 2: '
 3: e$=CHR$(27)
 4: d$="*********************"
 5: a$=e$+"f"+e$+"E"+e$+"Y"+CHR$(32+11)+CHR$(32+52)+e$+"p"+d$
 6: a$=a$+e$+"Y"+CHR$(32+12)+CHR$(32)
 7: a$=a$+"*          "+e$+"RxVT52"+e$+"S        *"
 8: a$=a$+e$+"Y"+CHR$(32+13)+CHR$(32+56)
 9: a$=a$+"* "+e$+"G"+e$+"j--------------------"
10: a$=a$+e$+"k"+CHR$(10)+"© 1988 M.Schumacher"+e$+"F *"
11: a$=a$+e$+"Y"+CHR$(32+14)+CHR$(32+4)+d$+e$+"q"
12: FOR i=1 TO 56*8
13:   a$=a$+e$+"Y"+CHR$(32+11)+CHR$(32)+e$+"U"
14:   IF EVEN(i)
15:     a$=a$+e$+"Y"+CHR$(32+12)+CHR$(32)+e$+"U"
16:     a$=a$+e$+"Y"+CHR$(32+13)+CHR$(32)+e$+"T"
17:   ENDIF
18:   a$=a$+e$+"Y"+CHR$(32+14)+CHR$(32)+e$+"T"
19: NEXT i
20: a$=a$+e$+"Y"+CHR$(32+24)+CHR$(32)
21: a$=a$+STRING$(14,e$+"Z")+e$+"a"
22: OPEN "O",#1,"demobild.dat"
23: PRINT #1,a$;
```

*Listing 5: "Testbild"-Generator für xVT52*

**ENDE**

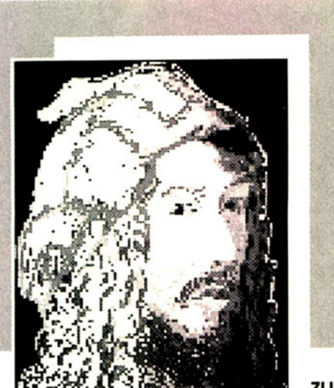



Demoversion anfordern! (DM 15,–)

philgerma
Barerstr. 32
8000 München 2
TEL. 089-281228




**das geniale Universal Programm**
für Atari ST



Malen
Zeichnen
Schreiben
Bewegen




DM 78,–

zum absoluten Niedrigpreis!

Demoversion anfordern! (DM 10,–)

philgerma
Barerstr. 32
8000 München 2
TEL. 089-281228

philgerma
**INFO**

**Unser Service endet nicht an der Ladentür. Auch bei Versandbestellung garantieren wir Ihnen unsere volle Unterstützung.**

## Sprachen

| | |
|---|---|
| **Lattice C** (MCC) | DM 298,– |
| **MCC Assembler** (Metacomco) | DM 168,– |
| **MCC Pascal 2** (Metacomo) | DM 248,– |
| **Pro Pascal** (Prospero) | DM 248,– |
| **Pro Fortran 77** (Prospero) | DM 378,– |
| **Modula-2 Standard** (TDI) | DM 248,– |
| **Modula-2 Developer** (TDI) | DM 398,– |
| **Omikron Basic Interpreter** | DM 178,– |
| **Omikron Basic Compiler** | DM 178,– |
| **AC Fortran 77** (absoft) | DM 448,– |
| **True Basic** Interp./Comp. | DM 248,– |
| **LDW Basic** Atari Bas. komp. | DM 98,– |
| **GFA Basic Interpreter** | DM 98,– |
| **GFA Basic Compiler** | DM 98,– |
| **UTAH Cobol + PC-Ditto V. 3.64** | DM 398,– |
| **OS9 Betriebssystem** mit Compilern für C, Basic, Pascal, Assembler + Tab. kalk, Textv., Datenbank | DM 1598,– |

## Text

| | |
|---|---|
| **Star Writer ST** dt. Textv. | DM 198,– |
| **1st Word plus** deutsch | DM 198,– |
| **1st proportional** | DM 88,– |

## Business

| | |
|---|---|
| **T.I.M.** kleine Buchhaltung | DM 298,– |
| **T.I.M.** Cashflow Kassenbuch | DM 298,– |
| **FibuMAN f** prof. Buchhaltung | DM 768,– |
| **K-Spread 2** Tab. kalk. | DM 228,– |
| **K-Graph 2** Grafik + Statistik | DM 148,– |
| **K-Comm 2** Terminalprogramm | DM 148,– |
| **dBMAN** Datenbank deutsch | DM 399,– |
| **BASiCALC** Tabellenkalk. dt. | DM 78,– |
| **AHJ-Backup** Harddisksicherung | DM 78,– |

## Spiele

| | |
|---|---|
| **Psion Chess** | DM 69,– |
| **Flight II** Flugsim. s/w u. f. | DM 119,– |
| **Bubble Bobble** | DM 59,– |
| **Star Trek** | DM 49,– |
| **Guild of Thieves** | DM 59,– |
| **Jinxter** | DM 69,– |

## Hardware

| | |
|---|---|
| **Einzellaufwerk 3.5"** 720 KB | DM 348,– |
| **Doppellaufwerk 3.5"** 2*720 KB | DM 648,– |
| **Einzellaufwerk, 5.25"** 40/80 Sp. | DM 448,– |
| **Doppellaufwerk, 3.5" + 5.25"** | DM 698,– |
| **K-MAX** Transputerboard | DM 2598,– |
| **Cascade OCCAM** Transputer-Software f. K-MAX | DM 598,– |
| **OMEGA Farbgrafikkarte** Auflösung 1024*512 | DM 2698,– |
| **10 Disketten 3.5"** 2DD no name | DM 29,– |

Bei Bestellungen unter DM 200,– beträgt der Versandkostenanteil DM 4,80. Nachnahme DM 3,20. Ins Ausland liefern wir nur gegen Vorkasse (Überweisung oder Euroscheck).

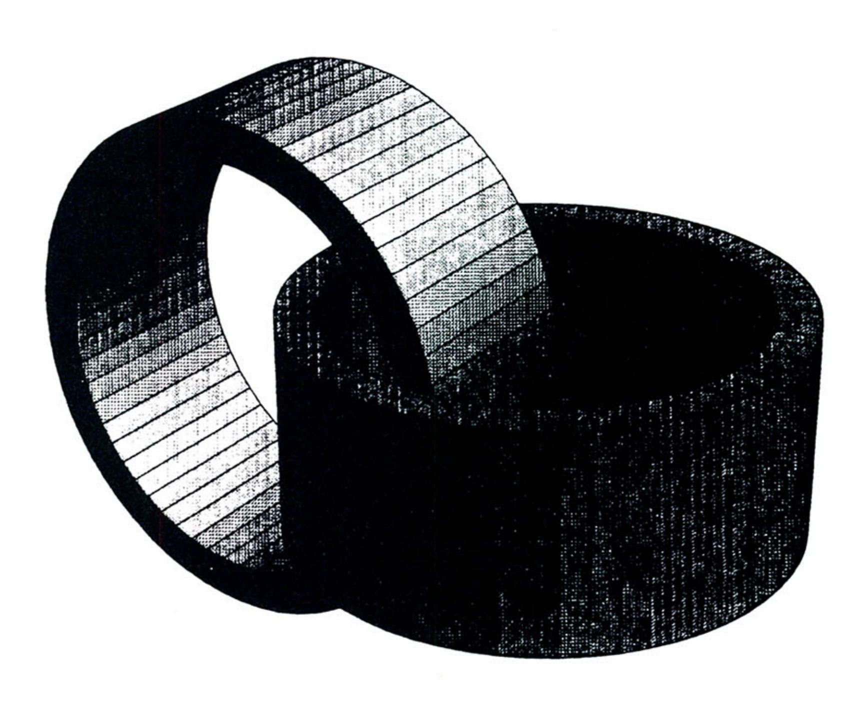
**Telefonische Bestellannahme und Hotline-Service: 089/281228**
von Mo – Fr 9.00 – 18.30 Uhr
Sa 10.00 – 14.00 Uhr

Preis- bzw. Händlerlisten anfordern bei

philgerma
Barerstr. 32
8000 München 2
TEL. 089-281228

# DIN A4-FUNKTIONS-PLOTTER

**Programme zur Darstellung dreidimensionaler Funktionen gibt es in großer Menge - aber nur die wenigsten arbeiten mit einer Auflösung von 2000*1440 Punkte. Das im folgenden besprochen Programm hat nicht nur diese Eigenschaft, es vermittelt zudem eine plastische Vorstellung der darzustellenden Funktion durch die Berücksichtigung der Helligkeitsverteilung auf der Oberfläche infolge einer (fiktiven) Lichtquelle.**



*Ein Beispiel für eine andere Anwendung des hier vorgestellten Druckerprinzipes*

Das Programm arbeitet im hochauflösenden Modus des Atari ST und ermöglicht die Ausgabe der dreidimensionalen Grafik im DIN A4 Format am NEC P6 (und ähnlichen 24 Nadel-Druckern). Bei der Besprechung des Programmes wird zuerst auf die dreidimensionale Darstellung einer gegebenen Funktion am Bildschirm eingegangen. Anschließend wird gezeigt, wie diese Grafik in hoher Auflösung relativ einfach gedruckt werden kann.

## Überblick

Um die Funktion zu zeichnen, wird sie in zahllose kleine Dreiecke zerlegt. Zu jedem Punkt dieser Dreiecke wird aus den X- und Y-Koordinaten mit der darzustellenden mathematischen Funktion die dazugehörige Z-Koordinate berechnet. Jeweils 3 solche Koordinatenpunkte bilden ein kleines Dreick.

Damit die Funktion in richtiger Sichtbarkeit dargestellt wird, werden die Dreiecke Zeile um Zeile von hinten beginnend gezeichnet - wenn die Fläche wegen einer Wölbung weiter hinten befindliche Teile verdeckt, dann werden diese ohne aufwendige Untersuchung einfach durch die neuen Dreiecke überdeckt und so unsichtbar. Um darüber hinaus einen plastischen Eindruck der Ebene zu vermitteln, wird der Einfallswinkel von Lichtstrahen auf die Fläche in Form von Grauschattierungen der Dreiecke berücksichtigt.

## Die Details

Im Programm werden zu Beginn einige Variablen mit Startwerten belegt: die Bedeutung von SCALE und LINIENSTAERKE wird etwas weiter unten besprochen. BASE bestimmt die Größe eines Dreiecks - je kleiner dieser Wert gewählt wird, desto genauer wird die Darstellung, desto länger die Rechenzeit und desto größer der Speicherbedarf. Der Beleuchtungsvektor gibt die Richtung an, unter der eine Lichtquelle in unendlich großer Entfernung (damit fallen die Lichtstrahlen parallel ein) Licht aussendet. Dieser Vektor wird durch seine Länge dividiert, sodaß sich (ohne Veränderung der Richtung) seine Länge auf 1 reduziert - dies ist zur Berechnung der Helligkeit am günstigsten.

Sodann wird der Bereich für X- und Y-Koordinaten festgelegt, in dem die Ebene dargestellt wird. Die Werte sind hier so gewählt, daß das letzte (rechte) Achtel des Bildschirms nicht genutzt wird - der Grund dafür liegt im gebräuchlichen Papier-Format, das nicht ganz den Proportionen des Bildschirms entspricht.

Nach einigen Feld-Dimensionierungen wird das Unterprogramm GRAUTON aufgerufen: darin werden aus DATA-Zeilen insgesamt 32 Grau-Schattierungen in ein String-Feld gelesen, wobei der Aufbau der Zeichenketten dergestalt ist, daß sie zur Definition eines eigenen Füllmusters mit DEFFILL verwendet werden können. Die Schattierungen werden durch die Besetzung von einer zunehmenden Anzahl von Punkten in einer 8*8 Matriz erzeugt. Diese Schattierungsmuster ermöglichen einen kontinuierlichen Farb-Übergang zwischen Weiß und Schwarz.

Es folgen nun recht kompliziert wirkende Funktions-Definitionen: mit ihnen wird aus gegebenen X- und Y-Koordinaten die Z-Koordinate der Funktion berechnet. Dazu verwendet die Hauptfunktion Z() eine Unterfunktion, die den Sinus-Wert einer rechnerischen Kombination der X- und Y-Koordinaten bestimmt. Durch die Überlagerung mehrerer solcher Funktionen durch Addition ergeben sich zahllose Möglichkeiten, interessante Flächen zu bilden. Der Wertebereich einer eigenen Funktions-Definition sollte zwischen +/- 35 liegen, damit die Bildschirmgrenzen nicht überschritten werden. Zwei weitere Beispiele für Funktionsdefinitionen (für Abb. 5 und 6):

```
Deffn Z(X,Y)=Sin((X-250)*(Y-160)/
        2000)*20+Sin((X+Y*2)/30)*8
Deffn Z(X,Y)=@Uf1(X-250,Y150)
        *10+@Uf2(X-200,Y-200)*8
Deffn Uf1(A,B)=Sin(Sqr(A^2+B^2)/10)
Deffn Uf2(A,B)=Sin((A+2*B) 20)
        +Sin(Sqr(A^2+B^2)/16)
        *1.4+Sin(A*B/3000)
```

Es sollte noch erwähnt werden, daß das verwendete Koordinatensystem ein 'Links-System' ist (Abb. 1), während in der Mathematik gewöhnlich ein 'Rechts-System' verwendet wird, bei dem die Y-Koordinate in die entgegengesetzte Richtung zeigt. Wenn es also von Bedeutung ist, daß die Funktion nicht spiegelverkehrt dargestellt wird, dann muß bei der Funktionsdefinition die Y-Koordinate jeweils mit negativem Vorzeichen verwendet werden.

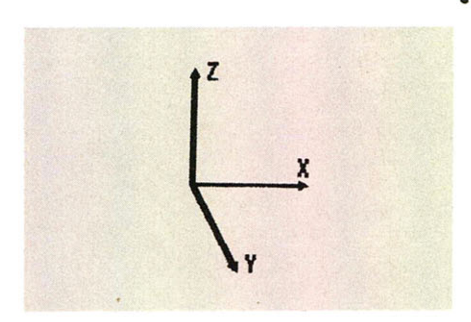



*Bild 1: Das verwendete Koordinaten-system*

In zwei einfachen Schleifen werden sodann die Z-Koordinaten zu Punkten in einem quadratischen XY-Raster berechnet und in einem Feld gespeichert (um Rechenzeit zu sparen). Die Subtraktion von 35 gewährleistet, daß die Dreiecks-Flächen nicht über den oberen Bildschirmrand hinausragen. Wie sich aus diesen Koordinatenwerten Dreiecke ergeben, wird in Abb. 2 gezeigt.

Es folgt eine weitere geschachtelte Schleife, in der die Dreiecke zeilenweise am Bildschirm ausgegeben werden. Dabei wird die Helligkeit der einzelnen Flächen mit der HELLIGKEIT-Funktion bestimmt: diese Funktion ermittelt den Cosinus-Wert des Winkels zwischen dem Beleuchtungsvektor und dem Normalvektor auf ein einzelnes Dreieck. Dieser Wert wird durch die Bildung des skalaren Produktes und anschließender Division durch die Länge des Normalvektors gebildet. Der Wert ist ein Maß für die Helligkeit der Fläche [1].

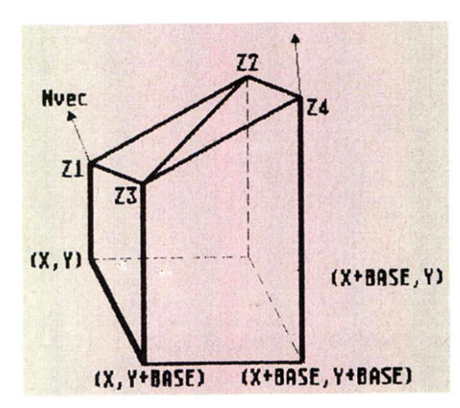



*Bild 2: Die Fläche wird durch viele Dreiecke dieser Art angenähert*

Jetzt muß nur noch geklärt werden, wie der Normalvektor auf eine Dreiecksfläche gebildet wird. Die Vorgangsweise wird am Beispiel des linken der zwei Dreiecke in Abb.2 demonstriert: es werden dabei die zwei Vektoren zwischen den Punkten mit der Höhe Z1 und Z2 bzw. Z1 und Z3 verwendet, die folgendermaßen aussehen:

```
      {BASE }             {0     }
vek13= {0     }     vek12= {BASE }
      {Z2-Z1 }             {Z3-Z1 }
```

Der Normalvektor, der auch in Abb.2 zu sehen ist('Nvec'), ergibt sich durch kreuzweises Multiplizieren der Komponenten dieser Vektoren:

```
              { 0-BASE*(Z2-Z1) }            {Z1-Z2}
vek13 x vek12 = { 0-BASE*(Z3-Z1) } = BASE *  {Z1-Z3}
              { BASE*BASE-0    }            {BASE }
```

Mit diesem Vektor wird die Helligkeitsfunktion aufgerufen, wobei der gemeinsame Faktor BASE nicht berücksichtigt wird. In analoger Weise wird auch die Helligkeit des anderen Dreiecks berechnet. Um die Farbe bei der Ausgabe am Drucker nicht nochmals berechnen zu müssen, wird sie ebenfalls in Feldern gespeichert.

## Ein Dreieck zeichnen

... ist gar nicht so einfach: dem Unterprogramm Dreieck werden die dreidimensionalen Koordinaten der drei Eckpunkte übergeben. Daraus werden zweidimensionale Bildschirmkoordinaten berechnet. Zu X wird ein Drittel von Y addiert, damit die Fläche schräg im Raum steht. Außerdem wird Z von Y subtrahiert (weil sonst wegen des Bildschirm-Koordinatensystems positive Z-Werte nach unten gezeichnet würden). Diese Berechnungsweise entspricht im wesentlichen einer etwas vereinfachten Zentral-Projektion. Darüber hinaus werden sämtliche Koordinatenwerte mit SCALE multipliziert. Dieser Faktor ist für die Ausgabe am Bildschirm 1, für die Ausgabe am Drucker wird er später auf 3.6 erhöht.

Es folgt eine nicht ganz leicht verständliche Schleife, in der nach identischen Y-Koordinaten von je zwei Eckpunkten des Dreiecks gesucht wird. Falls solche Punkte existieren, werden die Koordinaten um 1 verkleinert, wenn sie kleiner als der Y-Wert des dritten Punktes sind, andernfalls um 1 vergrößert. Der Grund für diese komplizierte Prozedur liegt in einem Fehler der POLYFILL Routine (TOS von Diskette): dieser Fehler wird nur dann bemerkbar, wenn die Flächen ohne Umrandung gezeichnet werden. Es treten dann gelegentlich weiße Streifen zwischen zwei aneinanderliegenden Flächen auf.

Insgesamt gibt es drei Möglichkeiten, die Flächen zu zeichnen: ohne Rand (LINIENSTAERKE=0), mit dünnem Rand (L=1) und mit etwas dickerem Rand (L=2). Die Abbildungen 4 bis 6 zeigen die sich dadurch ergebenden Grafiken. Gewöhnlich werden die Dreiecke mit einem ihrer Helligkeit entsprechendem Muster angefüllt, es ist aber natürlich auch möglich, alle Flächen einfach weiß anzufüllen - dazu wird die Funktion HELLIGKEIT als konstant 1 definiert:

```
Deffn Helligkeit (X,Y,Z)=1
```

Der beste plastische Eindruck ergibt sich, wenn auf eine Umrandung der Flächen verzichtet wird, Details der Gestalt der Fläche werden aber besser sichtbar, wenn die Dreiecke zwar umrandet, aber dafür ohne Graustufen dargestellt werden (besonders bei sehr komplizierten Funktionen ratsam).

Noch kurz einige Worte zur Rechenzeit: sie ist sehr klein, wenn die Funktion nur am Bildschirm (und nicht am Drucker) gezeichnet wird - bei BASE=5 (relativ

hohe Auflösung - die Funktion wird durch ~12000 Dreiecke angenähert) dauert dieser Vorgang im compilierten Programm keine zwei Minuten.

## Die Ausgabe am Drucker

Der einfachste Weg, die bisher erzeugte Grafik zu Papier zu bringen wäre eine direkte Hardcopy; dabei würde aber die Auflösung unverändert bleiben, was ein entscheidender Nachteil ist - der NEC P6 kann auf einer DIN A4 Seite fast 3 Millionen Punkte (in 180 DPI) darstellen - das ist etwa das 12 fache der Auflösung des Atari SW-Monitors. Es wird daher ein Weg gesucht, diese hohe Auflösung möglichst einfach zu nutzen.

Die Funktion wird am Drucker längs ausgedruckt (damit hat der Ausdruck ein ähnliches Format wie der Bildschirm). Sie wird dazu um den Faktor 3.6 vergrößert - damit werden aus 400 vertikalen Punkten am Bildschirm 1440 Punkte horizontal am Drucker, womit die gesamte Breite des Druckers genutzt wird.

Um die Grafik in der hohen Auflösung zeichnen zu können, wird sie in Segmente zerlegt, die am Atari Bildschirm dargestellt werden können. Die Grafik wird in 4 Streifen geteilt, die nacheinander gedruckt werden. Jeder dieser Streifen besteht wiederum aus 4 Segmenten (siehe Abb. 3).

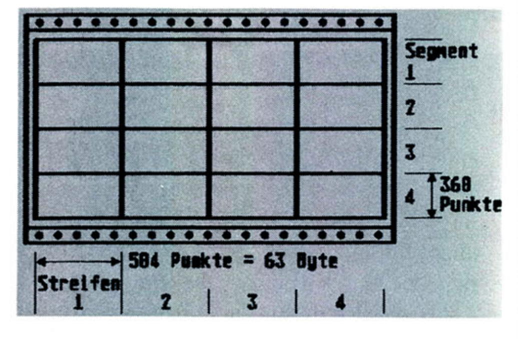



***Abbildung 3: auf einem Endlos-Blatt wird schematisch die Aufteilung der hochauflösenden Grafik in Segmente gezeigt***

Dieser Zerstückelungs-Vorgang wird vom Unterprogramm NEC_PRINT gesteuert. Die Größe eines Segmentes ist 360 Punkte vertikal (360*4=1440) und 63 Bytes=504 Punkte horizontal (die Byteanzahl muß ein Vielfaches von 3 sein, da beim NEC-Drucker immer 24 Punkte (=3 Byte) zugleich gedruckt werden). Mit 4 Streifen ergeben sich am Drucker vertikal 2016 Punkte - damit wird eine DIN A4 Seite weitgehend gefüllt.

```
 1:  ' PLOTTER - dreidimensionale Funktionen am Bildschirm darstellen und
 2:  '           wahlweise im DIN A4 Format am NEC P6 ausdrucken
 3:  '
 4:  ' Variablen belegen, Felder dimensionieren
 5:  '
 6:  Scale=1                              !Vergrößerungsfaktor für Ausgabe
 7:  Linienstaerke=1                      !Linienstärke der Umrandung der Dreiecke
 8:  Dpoke Intin,Sgn(Linienstaerke)       !Umrandung bei POLYFILL ein/aus
 9:  Dpoke Contrl+2,0                     ! mit der VDI-Funktion Set_Fill_
10:  Dpoke Contrl+6,1                     ! Perimeter_Visibility
11:  Vdisys 104
12:  Base=10                              !Basislänge einer einzelnen Dreiecksfläche
13:  Lx=5                                 !Beleuchtungsvektor: zeigt vom unter-
14:  Ly=-4                                ! suchten Punkt in Richtung einer
15:  Lz=9                                 ! unendlich weit entfernten Lichtquelle
16:  Llen=Sqr(Lx^2+Ly^2+Lz^2)             !diesen Vektor auf Länge 1 verkürzen
17:  Div Lx,Llen
18:  Div Ly,Llen
19:  Div Lz,Llen
20:  Xreso=445                            !Koord.bereich der dargestellten Ebene
21:  Yreso=330
22:  Line 560,0,560,400                   !bis zu dieser Linie wird Grafik gedruckt
23:  Dim X%(3),Y%(3)                      !Feld zum Zeichen der Dreiecke
24:  Dim Z(Xreso/Base,Yreso/Base)             !Feld für Funktionswerte der Fläche
25:  Dim Farbe%(Xreso/Base,Yreso/Base,1)  !Feld für Farben der einzelnen Dreiecke
26:  Dim Grauton$(32)                         !Feld für Graustufen-Muster
27:  @Grauton                                 !die Daten für Schattierungen einlesen
28:  '
29:  ' Funktion und Unterfunktionen, die die dreidimensionale Ebene bilden
30:  Deffn Z(X,Y)=@Uf(X-250,Y-180)*15+@Uf(X-170,Y-80)*15
31:  Deffn Uf(A,B)=Sin(Sqr(A^2+B^2)/16)
32:  '
33:  ' die Funkionswerte der Fläche erechnen und in Feld speichern
34:  For Y=0 To Yreso Step Base
35:    For X=0 To Xreso Step Base
36:      Z(X/Base,Y/Base)=@Z(X,Y)-35    !vom Funktionswert 35 subtrahieren,
37:    Next X                           ! damit Funktion nicht über den oberen
38:  Next Y                             ! Bildschirmrand hinausragt
39:  '
40:  ' Funktion zur Helligkeits-Berechnung eines Dreiecks
41:  Deffn Helligkeit(X,Y,Z)=(Lx*X+Ly*Y+Lz*Z)/Sqr(X^2+Y^2+Z^2)
42:  '
43:  ' Fläche am Bildschirm ausgeben, dabei Farbe der Dreiecke speichern
44:  For Y=0 To Yreso-Base Step Base
45:    For X=0 To Xreso-Base Step Base
46:      X%=X/Base                        !Index auf Feld für Funktionswerte
47:      Y%=Y/Base
48:      Z1=Z(X%,Y%)                      !die 4 Funktionswerte ergeben die
49:      Z2=Z(X%+1,Y%)                    ! Z-Koordinaten für die Eckpunkte
50:      Z3=Z(X%,Y%+1)                    ! zweier Dreiecksflächen
51:      Z4=Z(X%+1,Y%+1)
52:      Farbe1=Max(@Helligkeit(Z1-Z2,Z1-Z3,Base)*Farbanzahl,0)
53:      Farbe2=Max(@Helligkeit(Z3-Z4,Z2-Z4,Base)*Farbanzahl,0)
54:      Farbe%(X%,Y%,0)=Farbe1         !Farbe der Flächen in Feld speichern
55:      Farbe%(X%,Y%,1)=Farbe2
56:      @Dreieck(X,X+Base,X,Y,Y,Y+Base,Z1,Z2,Z3,Farbe1)
57:      @Dreieck(X,X+Base,X+Base,Y+Base,Y+Base,Y,Z3,Z4,Z2,Farbe2)
58:    Next X
59:  Next Y
60:  '
61:  ' Ausgabe der Grafik am Drucker
62:  Print "Ausdruck starten ? (J/N)"
63:  A=Inp(2)
64:  If A=Asc("j") Or A=Asc("J")
65:    Scale=3.6                       !Vergrößerungsfaktor für Ausgabe am NEC P6
66:    @Nec_print
67:  Endif
68:  '
69:  Procedure Dreieck(X1,X2,X3,Y1,Y2,Y3,Z1,Z2,Z3,Farbe%)
70:    ' Unterprogramm, das aus den dreidimensionalen Eckpunktkoordinaten einer
71:    ' dreieckigen Fläche diese am Bildschirm (zweidimensional) darstellt
72:    X%(0)=(X1+Y1/3)*Scale     !Umrechnung in zweidimensionale Koordinaten:
73:    X%(1)=(X2+Y2/3)*Scale     ! zur X-Koordinate wird ein Drittel der
74:    X%(2)=(X3+Y3/3)*Scale     ! Y-Koordinate addiert - dadurch kommt die
75:    Y%(0)=(Y1-Z1)*Scale       ! Schrägstellung der Fläche zustande; die
76:    Y%(1)=(Y2-Z2)*Scale       ! Y-Koord. wird um die Z-Koord. verkleinert;
77:    Y%(2)=(Y3-Z3)*Scale       ! alle Koord. werden Scale multipliziert
78:    If Linienstaerke=0
79:      For A%=0 To 2           !Fehler der POLYFILL-Routine auffangen: wenn
80:        B%=(A%+1) Mod 3       ! ein Dreieck gezeichnet werden soll, bei dem
81:        C%=(A%+2) Mod 3       ! eine Linie horizontal verläuft, so wird
82:        If Y%(A%)=Y%(B%)      ! diese horizontale Linie nicht mit dem Muster
83:          If Y%(A%)<Y%(C%)    ! angefüllt; deshalb werden in diesem Fall die
84:            Dec Y%(A%)        ! entsprechenden Y-Koord.werte um 1 erhöht
85:            Dec Y%(B%)        ! oder vermindert; dieser Fehler ist nur dann
86:          Else                ! bemerkbar, wenn die Routine ohne Umrandung
87:            Inc Y%(A%)        ! der zu zeichnenden Fläche verwendet wird
88:            Inc Y%(B%)
89:          Endif
90:        Endif
91:      Next A%
92:    Endif
93:    X%(3)=X%(0)               !für Umrandung mit POLYLINE muß ein
94:    Y%(3)=Y%(0)               ! geschlossener Linienzug vorhanden sein
```

## Alexander-Computer

**Beratung, Service und Verkauf**
**Atari, Brother, Commodore, Epson, HP, NEC, Star, Schneider**

```
*******        SH-205    989.00  DM          *******
1040 STF, komplett mit Monitor SM 124 ...1469.-
STFM, 512KB, Maus, Laufwerk, Tos-Roms ....729.-
SM124 Monitor ...439.-    Vortex HD20 ...1098.-
Mega ST-2/ST-4 ....................auf Anfrage
Atari Laserdrucker komplett ........auf Anfrage
in Kürze Atari PC 2 / PC 3 .........auf Anfrage
NEC P6 .....1060.-  NEC P2200 ...........889.-
STAR LC-10 ...575.-  Star LC-10 Color  ...698.-
STAR NB24-10 24-Nadeln + Traktor ........1198.-
LQ-500 .....869.-   Brother M 1724L .....1495.-
Weitere Modelle sowie Software auf Anfrage.
Fordern Sie unsere neuste Hard- und Software-
liste an.
```

Alexander-Computer,5030Hürth,Postfach1111,TEL.02233/63329

## BUSCH & REMPE DATENTECHNIK

**ATARI 1040 STF** mit Monitor **SM 124** — *1430,-- DM*

**DISKETTENLAUFWERKE** für ATARI ST oder AMIGA 2•80 Track vollkompatibel
Qualitätslaufwerke TEAC anschlußfertig im Metallgehäuse **3.5"** — *259,-- DM*
**5.25" incl. 40/80 Tr. Umschaltung** — *349,-- DM*
TEAC Diskettenlaufwerke 2•80 Spuren, 3.5" FD135-FN *189,-- DM*, 5.25" FD55-FR *229,— DM*

**AT-TASTATUR** für alle **ATARI ST**
komlett mit AT-Tastatur, Interface und deutscher Einbauanleitung — *498,-- DM*

**Lightspeed 1200 vollduplex MODEM** — *349,-- DM*
2400 Baud Version 698,-- DM Modems auch mit CCITT V 23 (BTX) lieferbar
Der Betrieb der Modems ist am öffentlichen Telefonnetz der Bundespost innerhalb der BRD und Berlin-West nicht gestattet.
Bitte beachten Sie die Bestimmungen des Fernmeldeanlagengesetzes

**MAILBOXPROGRAMM-ZERBERUS** — *298,-- DM*
nach Geonet Standart, befehlsorientiert, selbstabrechnend, Editor, voll fernbedienbar, X-Modem uvm.

**STAR LC 10** kompat. Drucker mit Papierparker 4k Puffer NLQ — *528,-- DM*

**Lützowstr. 98 4200 Oberhausen 11** ☎ 0208/687886 0203/478256

## ☆☆☆ ATARI ST ☆☆☆

**Anwendersoftware**

| | |
|---|---|
| **BS-Handel** Vers. 2.0 | 498,-- |
| **BS-Fibu** Vers. 2..0 | 598,-- |
| 1st Word Plus | 189,-- |
| Signum Zwei | 399,-- |
| Publishing Partner | 249,-- |
| Superbase | 249,-- |
| dB-MAN | 398,-- |
| Adimens ST | 189,-- |
| **BTX-Manager** Vers. 2.0 | 428,-- |

**Sprachen/Entwicklung/Grafik**

| | |
|---|---|
| **GFA Basic** Vers. 2.0 | 89,-- |
| **GFA Basic** Vers. 3.0 | 189,-- |
| GFA Compiler | 89,-- |
| **GFA Draft plus** CAD Prog. | 329,-- |
| **Lattice C-Compiler** V. 3.04 | 289,-- |
| Megamax C-Compiler | 398,-- |
| colorSTar | 79,-- |
| Film Director | 139,-- |
| Degas Elite | 179,-- |

**Spiele**

| | |
|---|---|
| Bad Cat | 54,-- |
| Barbarian (Psygnosis) | 64,-- |
| Bard's Tale | 79,-- |
| Black Lamp | 57,-- |
| Bolo | 62,-- |
| Bureaucracy | 89,-- |
| Championship Wrestling | 49,-- |
| Defender of the Crown | 74,-- |
| Deja Vu | 64,-- |
| Dungeon Master | 72,-- |
| Fire Blaster | 54,-- |
| Flight Simulator II | 119,-- |
| Gauntlet II | 69,-- |
| Giana Sisters | 54,-- |
| Goldrunner II | 56,-- |
| Gunship | 72,-- |
| Hellowoon | 59,-- |
| Indiana Jones | 52,-- |
| Jagd auf Roter Oktober | 72,-- |
| Jinxter | 72,-- |
| Kaiser | 119,-- |
| Karate Kid II | 59,-- |
| Kings Quest II u. III je | 59,-- |
| Marble Madness | 79,-- |
| Metrocross | 64,-- |
| Mission Elevator | 57,-- |
| Oids | 56,-- |
| Passengers on the Wind | 67,-- |
| Phantasie III | 64,-- |
| Psion Schach (deutsch) | 69,-- |
| Road Runner | 64,-- |
| Sentinel | 57,-- |
| Shadowgate | 72,-- |
| Shuttle II | 64,-- |
| Solomons Key | 52,-- |
| Starglider | 63,-- |
| Star Trek | 57,-- |
| Tanglewood | 52,-- |
| Tass Time in Tonetown | 69,-- |
| Terrorpods | 64,-- |
| Test Drive | 79,-- |
| The Guild of Thieves | 69,-- |
| The Pawn | 69,-- |
| Ultima II | 69,-- |
| Ultima III | 69,-- |
| Ultima IV | 69,-- |
| Vermeer | 74,-- |

**Drucker**

| | |
|---|---|
| NEC P 2200 | 1049,-- |
| Epson LQ-500 | 998,-- |
| Epson LX-800 | 699,-- |
| Star LC 10 | 649,-- |

☞ Sofort kostenlos Preisliste bei Abteilung ST anfordern! ☜

**☞ Computer & Zubehör Versand ☜**
**Gerhard und Bernd Waller GbR**
**Kieler Str. 623, 2000 Hamburg 54, ☎ 040/5706007 + 5705275**

# Der König ist tot -

LOGICOM verabschiedet sich von seinen Kunden. Denn sie ermöglichten unser schnelles Wachstum.

Damit wir auch in Zukunft mit der bewährten Leistungsstärke für Sie da sind, heißt es ab sofort...

# es lebe der König!

LOGITEAM begrüßt die Kunden von LOGICOM, für die sich eigentlich nichts ändert.

Es sei denn, die neuen, größeren Räume, die bessere telefonische Erreichbarkeit, die längeren Öffnungszeiten - deshalb LOGITEAM.

**Unsere Angebote:**

## HARDWARE

| | |
|---|---|
| Atari 1040 STF (komplett) | DM 1.498,- |
| Atari Mega ST 2/4 | auf Anfrage |
| vortex HD plus 20 | DM 1.149,- |
| vortex HD plus 30 | DM 1.398,- |
| star LC 10 | DM 598,- |
| Epson LQ 500 | DM 998,- |
| NEC P 6 plus | auf Anfrage |
| NEC P 2200 | DM 998,- |

Alle Drucker mit deutschem Handbuch und Kabel.

**Weitere Hardware auf Anfrage.**

## SOFTWARE

Sämtliche Standard-Software lieferbar.

z.B. Application Systems, C.A.S.H., CCD, G DATA, GfA, GST, Hyper-Soft, u.v.m.

## Neu bei LOGITEAM:

Arzt-Software: Aus der Praxis für die Praxis

| | |
|---|---|
| Privatliquidation | DM 375,- |
| Praxisbuchhaltung | DM 375,- |

Hardware
Software
Beratung
Verkauf
Service




Kölner Str. 132 • 5210 Troisdorf • Tel.02241/71897-98

Der Druck eines Streifens erfolgt auf folgende Weise: es werden mit dem Unterprogramm PLOT_SEGMENT 4 Segmente mit unterschiedlichem Koordinaten-Ursprung gezeichnet. Der Koordinatenursprung kann problemlos durch Veränderung zweier Werte im WINDTAB Feld, das Daten für den Bildschirmaufbau enthält, verlegt werden. Damit ist das Unterprogramm leicht nachzuvollziehen. Es hat nur einen Nachteil: es ist relativ langsam, weil ja ein Großteil der gezeichneten Dreiecke außerhalb des am Bildschirm sichtbaren Segementes liegen.

Jedes der vier so gezeichneten Segmente wird in einer String-Variable durch GET gespeichert. Zu beachten ist hierbei nur, daß die obersten zwei Bildschirmzeilen nicht verwendet werden dürfen - dort treten ebenfalls Fehler der POLYFILL-Routine auf, wenn Polygone über den oberen Bildschirmrand hinausragen.

Sind die 4 Segmente somit in Zeichenketten gesichert, dann werden sie im Unterprogramm HARDCOPY gedruckt. Der Ausdruck erfolgt zeilenweise, wobei vertikale Streifen des Bildschirms am Drucker horizontal ausgedruckt werden. Da das Datenformat für die NEC Drucker wegen des 24-Nadel-Kopfes einen sehr einfachen Aufbau hat (3 Byte ergeben die 24 Druckelemente), können die Daten direkt mit PEEK vom Bildschirm gelesen werden, was natürlich der einfachste und schnellste Weg ist.

Die 1440 Punkte ergeben sich nun dadurch, daß zuerst 360 Spalten zu 24 Punkten aus dem 4. Segement gedruckt werden, dann 360 Spalten aus dem 3. Segment,.. Dieser Segmentwechsel ist sehr leicht zu realisieren, weil die Segmente nun mit PUT sehr schnell am Bildschirm projeziert werden können. Wichtig ist, daß die Segmente in verkehrter Reihenfolge und auch innerhalb der Segmente die Daten von unten nach oben gelesen und gedruckt werden - andernfalls wäre der Ausdruck spiegelverkehrt!

Dieser Vorgang wiederholt sich vier mal bis die gesamte Grafik am Papier ist - der Vorgang dauert mit einem compilierten Programm bei BASE=5 etwa 20 Minuten.

Noch einige Details zur Druckersteuerung: es wird mit 180 Dots per Inch (DPI) gedruckt. Dies ist nicht die höchste Auflösung des P6, ein Arbeiten mit 360 DPI wäre aber sinnlos, weil der Abstand zweier Punkte viel kleiner als die Größe eines einzelnen Punktes wäre. Auch bei 180 DPI wird eine Fläche, bei der nur jeder 2. Punkt schwarz ist, am Papier als

```
 95:     Deffill ,Grauton$(Farbe%) !Farbe% bestimmt den Grauton
 96:     Polyfill 3,X%(),Y%()      !Fläche zeichnen
 97:     If Linienstaerke=2                  !gegebenenfalls auch dickere Umrandung
 98:       Polyline 4,X%(),Y%() Offset 0,1 ! zeichnen (durch horizontale und
 99:       Polyline 4,X%(),Y%() Offset 1,0 ! vertikale Versetzung des Linien-
100:       Polyline 4,X%(),Y%() Offset 1,1 ! zuges)
101:     Endif
102:   Return
103:   ' ################## Ausgabe am Drucker
104:   Procedure Nec_print
105:     ' Haupt-Unterprog. zur großformatigen Ausgabe am NEC P6
106:     ' Zeilenabstand: nur 47/360 Zoll, um horizontale Streifen zu vermeiden
107:     Lprint Chr$(28)+Chr$(51)+Chr$(47);
108:     Bildanzahl=4          !Zahl der Bildsegmente
109:     Dim Bild$(Bildanzahl) !Felder für Bild-Daten
110:     Hoehe=360             !Zahl der Punkte (vertikal) pro Bildsegment
111:     Breite=63             !Zahl der Bytes (horizontal) pro Bildsegment
112:     Hidem                 !Maus stört nur
113:     For Streifen=0 To 3   !4 Streifen erfassen 63*8*4=2016 Punkte
114:       For Bild=0 To Bildanzahl-1                      !pro Streifen vier
115:         @Plot_segment(Streifen*Breite*8,Hoehe*Bild)  ! Segmente zeichnen
116:         Get 0,2,Breite*8,2+Hoehe,Bild$(Bild+1)       ! und speichern
117:       Next Bild
118:       @Hardcopy(Breite,Hoehe,Bildanzahl)   !den gesamten Streifen drucken
119:     Next Streifen
120:     Showm
121:     Lprint Chr$(27)+Chr$(50)   !normaler Zeilenabstand P6
122:     Lpoke Windtab+64,0         !Koordinatenursprung wieder nach (0,0)
123:   Return
124:   '
125:   Procedure Plot_segment(Xoffset,Yoffset)
126:     ' Bildschirmsegement am Bildschirm zeichnen, in String speichern
127:     Cls
128:     Dpoke Windtab+64,-Xoffset           !Koordinatenursprung verlegen
129:     Dpoke Windtab+66,-Yoffset
130:     For Y=0 To Yreso-Base Step Base     !die dreidimensionale Fläche zeichnen,
131:       For X=0 To Xreso-Base Step Base   ! wobei die früher berechneten Farb-
132:         X%=X/Base                       ! Werte verwendet werden;
133:         Y%=Y/Base                       ! es wird die gesamte Fläche
134:         Z1=Z(X%,Y%)                     ! gezeichnet, durch die Verlegung
135:         Z2=Z(X%+1,Y%)                   ! des Koordinatenursprungs
136:         Z3=Z(X%,Y%+1)                   ! und den Vergrößerungsfaktor
137:         Z4=Z(X%+1,Y%+1)                 ! (scale) entstehen die
138:         Farbe1=Farbe%(X%,Y%,0)          ! gewünschten Segmente
139:         Farbe2=Farbe%(X%,Y%,1)
140:         @Dreieck(X,X+Base,X,Y,Y,Y+Base,Z1,Z2,Z3,Farbe1)
141:         @Dreieck(X,X+Base,X+Base,Y+Base,Y+Base,Y,Z3,Z4,Z2,Farbe2)
142:       Next X
143:     Next Y
144:   Return
145:   Procedure Hardcopy(Breite,Hoehe,Bildanzahl)
146:     ' einen horiz. Streifen ausdrucken (bestehend aus 4 Bild-Segmenten)
147:     ' Hoehe ...  Zahl der Punkte vertikal am Bildschirm
148:     ' Breite ... Zahl der zu übertragenden Bytes; muß Vielfaches von 3 sein
149:     Spalten=Hoehe*Bildanzahl                    !Druckspalten am Drucker (1440)
150:     Adr%=Xbios(2)                               !Adresse des Bildschirmspeichers
151:     Cls
152:     For X%=0 To Breite-1 Step 3                 !X-Koord. am Bildschirm (in Byte)
153:       Lprint Chr$(27)+Chr$(86)+Chr$(2);         !jede Zeile doppelt drucken
154:       ' Grafikmodus (180 DPI)
155:       Lprint Chr$(27)+Chr$(42)+Chr$(39)+Chr$(Spalten)+Chr$(Spalten\256);
156:       For Bild=Bildanzahl Downto 1              !die Bildsegmente der Reihe nach
157:         Put 0,0,Bild$(Bild)                     ! projezieren und eine Zeile zu
158:         For Y%=Hoehe*80-80 To 0 Step -80        ! 24 Punkten drucken;
159:           Out 0,Peek(Adr%+Y%+X%)                ! dabei die Daten direkt vom
160:           Out 0,Peek(Adr%+Y%+X%+1)              ! Bildschirm lesen (von unten
161:           Out 0,Peek(Adr%+Y%+X%+2)              ! oben, damit Ausdruck nicht
162:         Next Y%                                 ! spiegelverkehrt)
163:       Next Bild
164:       Lprint Chr$(13);                          !Wagenrücklauf
165:       Lprint Chr$(27)+Chr$(86)+Chr$(0)          !Daten-Ende für Mehrfach-Druck
166:     Next X%
167:   Return
168:   '
169:   Procedure Grauton
170:     Restore Grauton_data
171:     Farbanzahl=32
172:     Dim Bit%(7)                                  !Hilfsfeld das binär das 8*8
173:     For Grau_ton=Farbanzahl Downto 0             ! Punkte-Muster enthält, das aus
174:       Clr Bit$                                   ! den DATA Zeilen gelesen und
175:       For I=0 To 7                               ! anschließend in Zeichenketten
176:         Bit$=Bit$+Mki$(Bit%(I)+Bit%(I)*256)      ! umgewandelt wird, die für den
177:       Next I                                     ! DEFFILL Befehl geeignet sind;
178:       Grauton$(Grau_ton)=Bit$+Bit$               ! die DATA-Werte enthalten die
179:       Read A,B                                   ! Koordinaten jenes Punktes
180:       Bit%(A)=Bit%(A) Or 2^B                     ! innerhalb der 8*8 Matrix, die
181:     Next Grau_ton                                ! gesetzt werden
182:     Erase Bit%()                                 !Hilfsfeld wieder löschen
183:     Grauton_data:
184:     Data 0,0, 4,4, 5,1, 1,5, 2,2, 6,3, 6,6, 0,4, 2,7, 5,7, 0,3, 4,6
185:     Data 7,2, 3,5, 4,2, 1,1, 1,3, 0,7, 6,5, 7,0, 2,3, 3,1, 7,4, 5,6
186:     Data 6,1, 3,6, 5,4, 7,7, 1,6, 3,3, 4,0, 0,2, 0,0
187:   Return
```

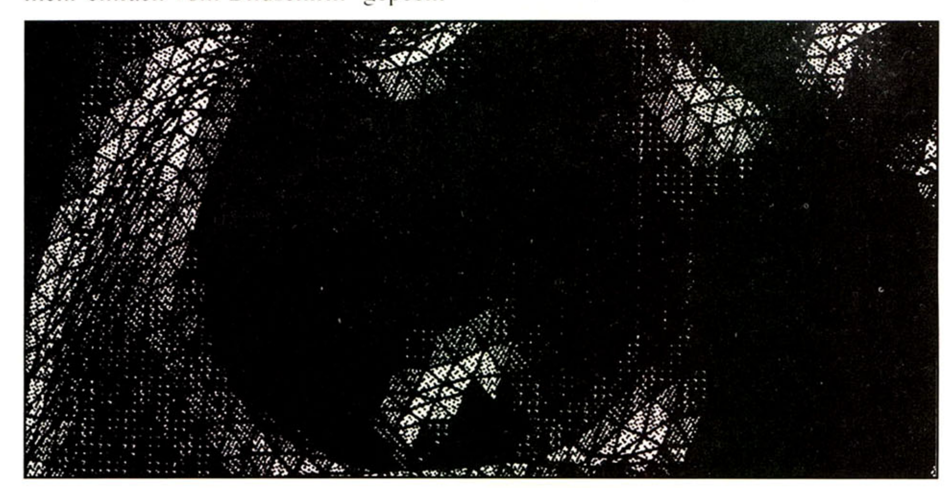
vollkommen schwarze Fläche dargestellt (weswegen die Grafiken am Bildschirm viel heller als am Drucker aussehen).
Jede Zeile wird doppelt gedruckt, um eine höhere Qualität (besonders bei etwas älteren Farbbändern) zu erreichen. Dies kann mit Steuercodes sehr einfach erreicht werden: [27 86 n] gibt den Beginn eines Datenblocks an, der n mal gedruckt werden soll, [27 86 0] das Ende dieses Datenblocks. Schon im Unterprogramm NECPRINT wird der Zeilenvorschub mit [28 51 47] auf 47/360 Zoll eingestellt - d.h. das sich je zwei Zeilen um 1/360 Zoll überlappen (eine Zeile hat die Breite 24/180 Zoll). Dadurch werden helle Streifen zwischen den Zeilen weitgehend vermieden, die auftretende Verzerrung des Bildes ist nicht wahrnehmbar.
Sollten die Druckergebnisse dennoch nicht zufriedenstellend sein, dann sollte überprüft werden, ob die Andruckstärke der Nadeln richtig eingestellt ist (siehe Handbuch des Druckers).
Wie aus der Beschreibung hervorgeht, müssen die vier Segmente zwischengespeichert werden, wofür etwa 100 KByte notwendig sind. Wenn dieser Speicher nicht vorhanden ist, dann kann die Breite der Segmentgröße verkleinert werden: es muß jedoch die Byte-Anzahl ein Vielfaches von 3 sein, weil ja immer 24=3*8 Punkte zugleich gedruckt werden. Je kleiner BREITE gewählt wird, desto größer ist die Zahl der notwendigen Streifen (und damit auch die Rechenzeit). Eine möglicher Wert wäre 42 - die Schleife für STREIFEN muß dann von 0 bis 5 laufen (Speicherbedarf ~ 70 Kbyte).
Das Programm läuft in der compilierten Version auch auf Minimalrechnern (512 KByte, Disketten-TOS) bis BASE=5; wer allerdings außerdem auf einen Compiler verzichten muß, muß die obige Verkleinerung der Segmente durchführen, damit das Programm läuft.

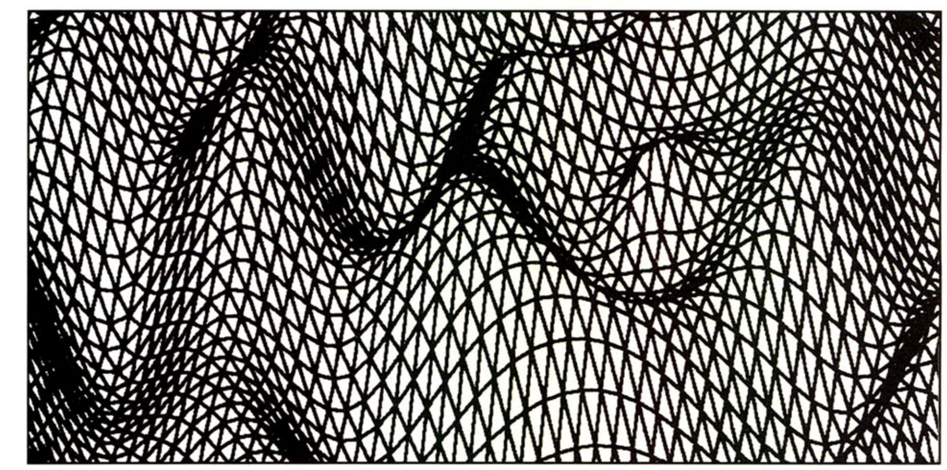
## Variationen

Die Idee dieses Druckverfahren ist grundsätzlich für jeden Computer und jeden Drucker (theoretisch auch für LASER-Drucker !) zu realisieren. Es ist allerdings nur für Grafiken geeignet, die einigermaßen rasch in beliebiger Vergrößerung vollständig gezeichnet werden können (das ist in der Regel nur für solche Grafiken der Fall, die durch eine begrenzte Zahl von Flächen oder Linien aufgebaut werden kann - z.B. das Resultat eines Hiddens-Surface Algorithmus).
Eine Grafik dieser Art wird durch Multiplikation der Bildschirm-Koordinaten vergrößert, in Segmenten gezeichnet, zwischengespeichert und sodann gedruckt. Ein wesentlicher Vorteil ist, daß dazu keine speziellen Zeichen-Routinen geschaffen werden brauchen - es muß lediglich der Koordinatenursprung verlegt werden (was natürlich auch rechnerisch für jeden einzelnen Punkt möglich wäre, wenn das Betriebssystem eine solche Möglichkeit nicht bietet).
In einem Punkt muß das Programm allerdings geändert werden, wenn ein Drucker mit 9 oder 18 Nadeln oder eine andere Auflösungstufe verwendet wird: es können dann die Daten für den Drucker nicht mehr einfach vom Bildschirm 'gepeekt' werden - sie müssen Punkt für Punkt durch POINT gelesen werden und rechnerisch zu Daten im dem Drucker genehmen Format umgewandelt werden, was zwar nicht kompliziert, aber relativ langsam ist. Die Größe der Segmente wird sich natürlich nach der Auflösung des verwendeten Druckers richten; abgesehen davon ist es aus Zeitgründen natürlich sinnvoll, die Segmente möglichst groß zu machen - Voraussetztung dazu ist aber ein ausreichender Speicher.

*Michael Kofler*

*Quellen:*
*[1] David F. Rogers: Procedural Elements For Computer Graphics, 1985 McGraw Hill*





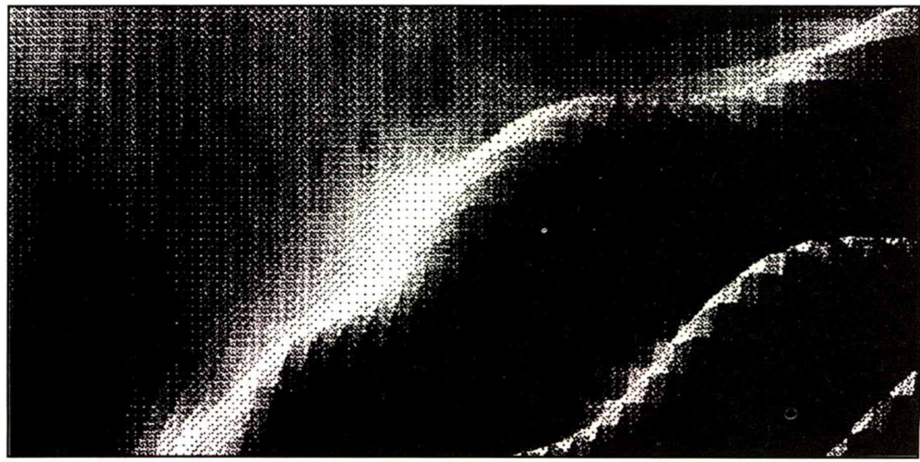
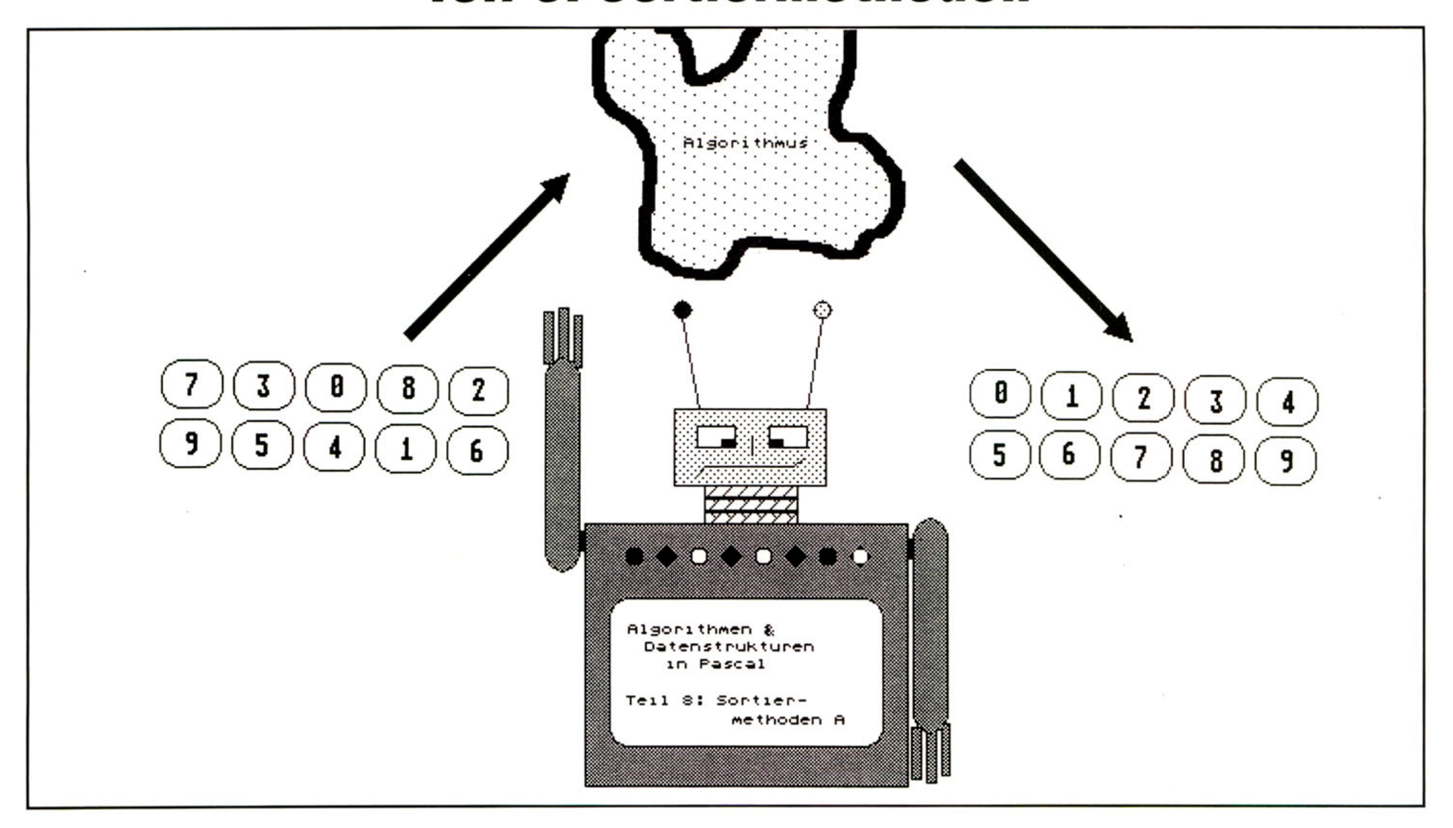


***Abbildung 4-6: Beispielausdrucke, die die 3 Arten zeigen, in der Grafiken dargestellt werden können.***

# ALGORITHMEN UND DATENSTRUKTUREN

## Teil 8: Sortiermethoden




**Nachdem ich nun die Datenverarbeitung glücklich hinter mich gebracht habe, möchte ich den letzten großen Abschnitt dieser Artikelserie einer Klasse von Algorithmen widmen, die praktisch für jede Art von Anwendung unentbehrlich sind. Ich spreche von Sortieralgorithmen. Dabei meine ich nicht die Art von Sortierung, die wir in den letzten Folgen durch Einordnung in irgendeine Art von Informationsstruktur erreicht haben, sondern die gleichzeitige Sortierung einer großen Anzahl von Elementen auf möglichst schnelle Weise. Wie Sie sehen werden, ist dabei der Weg vom einfachen Ansatz bis zu einem anspruchsvollen (schnellen) Programm weit.**

## Motivation

Die Sortierung von größeren Datenmengen ist eines der wenigen Gebiete in der Informatik, wo es sehr leicht möglich ist, direkte Vergleiche zwischen einzelnen Algorithmen in Form von Zeitmessungen anzustellen.
Diese quasi Wettkampfsituation hat eine große Anzahl von unterschiedlichen Algorithmen zu Tage gefördert, indem immer wieder irgendwelche Informatiker versuchten, die Algorithmen ihrer Konkurrenz zur Strecke zu bringen.
Um in etwa dieser Entwicklung Rechnung zu tragen, werde ich Ihnen in dieser und den zwei folgenden Teilen dieser Serie, beginnend mit den 'auf der Hand liegenden' Algorithmen bis hin zu den klassischen Sortieralgorithmen, eine kurze Zusammenfassung präsentieren.
Ich hoffe, daß ich Ihnen mit den dabei angestellten Zeitmessungen Impulse für DEN von Ihnen selbst zu entwickelnden Superalgorithmus, nennen wir ihn **hypersort**, geben kann. Leider muß ich gestehen, daß meine persönlichen Bemühungen nicht ausreichten den momentanen Spitzenreiter (**quicksort**) zu erreichen.

## Ansätze

Bei der Sortierung sind grundsätzlich immer zwei mögliche Ausgangssituationen zu beachten:
Entweder liegen die Daten in einer Filestruktur vor und müssen auf sequentielle Weise sortiert werden, oder wir haben eine Repräsentation in Arrays vorliegen und können so beliebig auf die Komponenten zugreifen.

Mit dem häufiger vorliegenden, letzteren Fall werden wir heute beginnen.
Die Repräsentation der Daten erfolgt in einer Type field, die im wesentlichen das Array a der Schlüssel enthält. Weiterer Bestandteil dieses field ist lediglich ein Zeiger auf das letzte Arrayelement (max), um auch variable Feldgrößen verarbeiten zu können.
Wir gehen dabei davon aus, daß der zu sortierende Anteil des Arrays im Bereich 1..max liegt. Die Hinzunahme der 0 in den Arraybereich ist rein programmtechnischer Natur und verdient daher keine weitere Beachtung. Die entsprechende Pascal-Notation finden Sie in den Zeilen 12-15 des Listings 8f.
Um nun einen solchen Array zu sortieren, unterscheidet man prinzipiell zwischen vier Methoden, die aber in den meisten Fällen keine ganz eindeutige Klassifizierung erlauben:

**1. Sortieren durch Einfügen**
Ein Element wird in eine schon sortierte Liste von Elementen eingefügt. Beginnend mit dem ersten Element des Arrays, was ja wohl zwangsläufig sortiert ist, wird nacheinander durch Einfügen der restlichen Array-Elemente eine sortierte Folge aufgebaut.

**2. Sortieren durch Auswählen**
Das Minimum des Arrays wird ausgewählt und an die erste Position gebracht. Durch wiederholte Anwendung dieses Vorgangs wird das gesamte Array sortiert.

**3. Sortieren durch Zerlegen**
Das Array wird in kleinere Arrays zerlegt, die nun einzeln sortiert werden.

**4. Sortieren durch Verschmelzen**
Zwei oder mehr sortierte Quellarrays werden miteinander verschmolzen, indem das Minimum der vordersten Arraypositionen in ein Zielarray kopiert wird. Durch Iterierung dieses Vorgangs entsteht ein größeres sortiertes Array. Diese Methode ist auch die einzige, die sich leicht auf Files übertragen läßt, da nur immer auf die erste Komponente zugegriffen wird. Doch dazu kommen wir später.

```
68 ¦ 77 28 35 95 58 40 2 99 82 51 78 49 61 31
68 77 ¦ 28 35 95 58 40 2 99 82 51 78 49 61 31
28 68 77 ¦ 35 95 58 40 2 99 82 51 78 49 61 31
28 35 68 77 ¦ 95 58 40 2 99 82 51 78 49 61 31
28 35 68 77 95 ¦ 58 40 2 99 82 51 78 49 61 31
28 35 58 68 77 95 ¦ 40 2 99 82 51 78 49 61 31
28 35 40 58 68 77 95 ¦ 2 99 82 51 78 49 61 31
2 28 35 40 58 68 77 95 ¦ 99 82 51 78 49 61 31
2 28 35 40 58 68 77 95 99 ¦ 82 51 78 49 61 31
2 28 35 40 58 68 77 82 95 99 ¦ 51 78 49 61 31
2 28 35 40 51 58 68 77 82 95 99 ¦ 78 49 61 31
2 28 35 40 51 58 68 77 78 82 95 99 ¦ 49 61 31
2 28 35 40 49 51 58 68 77 78 82 95 99 ¦ 61 31
2 28 35 40 49 51 58 61 68 77 78 82 95 99 ¦ 31
2 28 31 35 40 49 51 58 61 68 77 78 82 95 99 ¦
```

*Abb. 8a: Direktes Einfügen*

```
¦ 68 77 28 35 95 58 40 2 99 82 51 78 49 61 31
2 ¦ 77 28 35 95 58 40 68 99 82 51 78 49 61 31
2 28 ¦ 77 35 95 58 40 68 99 82 51 78 49 61 31
2 28 31 ¦ 35 95 58 40 68 99 82 51 78 49 61 77
2 28 31 35 ¦ 95 58 40 68 99 82 51 78 49 61 77
2 28 31 35 40 ¦ 58 95 68 99 82 51 78 49 61 77
2 28 31 35 40 49 ¦ 95 68 99 82 51 78 58 61 77
2 28 31 35 40 49 51 ¦ 68 99 82 95 78 58 61 77
2 28 31 35 40 49 51 58 ¦ 99 82 95 78 68 61 77
2 28 31 35 40 49 51 58 61 ¦ 82 95 78 68 99 77
2 28 31 35 40 49 51 58 61 68 ¦ 95 78 82 99 77
2 28 31 35 40 49 51 58 61 68 77 ¦ 78 82 99 95
2 28 31 35 40 49 51 58 61 68 77 78 ¦ 82 99 95
2 28 31 35 40 49 51 58 61 68 77 78 82 ¦ 99 95
2 28 31 35 40 49 51 58 61 68 77 78 82 95 99 ¦
```

*Abb. 8b: Direkte Auswahl*

## Direktes Einfügen

Zunächst betrachten wir einmal einen Algorithmus, der einer von den zweien 'auf der Hand liegenden' Algorithmen ist:
Das Sortieren durch direktes Einfügen.
Wir gehen davon aus, daß wir ein bereits sortiertes Array im Bereich 1..i-1 vorliegen haben und ein noch unsortiertes Teilstück im Bereich i..max. In diesem Fall fügen wir in den sortierten Bereich das Element a[i] ein, indem wir den sortierten Teilarray bis zu dem Punkt, wo seine Elemente kleiner werden als unser einzufügendes Element, um eine Position nach rechts schieben. Das Element a[i] wird dann in die frei werdende Stelle eingefügt. Diesen Algorithmus des 'direkten Einfügens' finden Sie im Listing 8a realisiert.
Die Abbildung 8a zeigt **direktes_einfuegen** bei der Arbeit. Dazu ist eine kleine Hilfsroutine zur Arrayausgabe notwendig, die man am Ende der FOR-Schleife in den Algorithmus einschmuggelt.
Beispielsweise: FOR l:=1 TO feld.max DO write(feld.a[l]);
Dadurch werden gewissermaßen Schnappschüsse des Arrays während der einzelnen Sortierphasen erstellt. Man sieht wie, von links beginnend, die sortierte Folge immer länger wird und schließlich das gesamte Array ausmacht.

## Direkte Auswahl

Der andere 'naheliegende' Algorithmus ist die direkte Auswahl des Minimums. Das Array wird dazu auf der Suche nach dem Minimum bis zum Ende durchlaufen. Die so ermittelte Stelle wird mit dem ersten Element vertauscht, und man erhält ein um ein Element vergrößertes, sortiertes Teilstück. Dieser Algorithmus wurde im Listing 8b realisiert. Wenn wir ihn auf die gleiche Folge wie oben anwenden, so ergeben sich die Arbeitsschritte der gleichnamigen Abbildung.

## Bubblesort #1

Ein Algorithmus mit recht lustigem Namen ist Bubblesort. Den Array von hinten durchlaufend, werden jeweils, wenn erforderlich, zwei benachbarte Elemente ausgetauscht. Diese Prozedur wird bis zum Array-Anfang durchgeführt, womit wir mindestens das Minimum an die erste Position gebracht haben, einige andere größere Elemente aber wahrscheinlich näher ans Ziel gerückt sind.
So erklärt sich auch der Name, indem man, zugegebenermaßen mit einiger

# SOFTWARE DIE ES IN SICH HAT!

**Nun auch in Deutschland!**

Schwarz auf Weiß kann jetzt jeder auf seinem ATARI ST unter MS-DOS arbeiten. Doch nicht nur monochrom, nein, auch in Farbe ist nun der Zugriff auf die Welt der PC-Rechner möglich.

**NEU EURO VERSION 3.64**

Die Software-Emulation **PC ditto** öffnet allen ATARI ST Anwendern das Tor zum gewohnten professionellen Business Standard.

Mit dem **PC ditto** können Sie mühelos mit Lotus 1-2-3 oder Symphony Ihre Kalkulation erstellen, oder Ihre Daten mit DBase III plus verwalten.

Mit dem **PC ditto** haben Sie Zugang zu Turbo Pascal, zum GW BASIC Interpreter, und auch zu Borlands neuestem Kind Turbo Basic.

Mit dem PC ditto läuft auch die Software, die es für den ATARI ST noch gar nicht gibt.

Mit dem **PC ditto** laufen so viele Programme, daß wir sie hier gar nicht alle auflisten können.

Der **PC ditto** unterstützt die ATARI-Festplatte, den Druckerport und alle Schnittstellen, soweit es die Hardware zuläßt.

Machen Sie aus Ihrem ATARI ST den preiswertesten PC-Clone! Bestellen Sie den **PC ditto** zum Superpreis von nur DM 198,-. **(Jetzt auch mit deutscher Anleitung)**

Wir laden Sie ein! Steigen Sie mit Ihrem ATARI ST in die Welt des MS-DOS ein.

Hiermit bestelle ich _ PC ditto für 198.-

Anruf genügt!

Telefon (0 61 96) 48 18 11, Mo-Fr 9-13 und 14-17 Uhr. Schriftliche Bestellung nur gegen Vorkasse oder Nachnahme (Versandkosten Inland DM 7,50, Ausland DM 10,—; bei Nachnahme zuzüglich DM 3,50 Nachnahmegebühr)

AUSLAND NUR GEGEN VORKASSE

Name:__________ Vorname__________

Straße:__________ Ort:__________

Unterschrift:__________

MS-DOS und GW-BASIC sind Warenzeichen von Microsoft Corp. Lotus 1-2-3+ Symphony sind Warenzeichen von Lotus Dev. Corp. Turbo Pascal und Turbo BASIC sind Warenzeichen von Borland Corp. DBase III Plus ist ein Warenzeichen von Ashton-Tate Corp.

Nähere Informationen gegen ausreichend frankierten Rückumschlag bei:




MERLIN COMPUTER GMBH
INDUSTRIESTRAßE 26
6236 ESCHBORN
TEL.: 06196/481811

```
68 77 28 35 95 58 40 2 99 82 51 78 49 61 31
 2 68 77 28 35 95 58 40 31 99 82 51 78 49 61
 2 28 68 77 31 35 95 58 40 49 99 82 51 78 61
 2 28 31 68 77 35 40 95 58 49 51 99 82 61 78
 2 28 31 35 68 77 40 49 95 58 51 61 99 82 78
 2 28 31 35 40 68 77 49 51 95 58 61 78 99 82
 2 28 31 35 40 49 68 77 51 58 95 61 78 82 99
 2 28 31 35 40 49 51 68 77 58 61 95 78 82 99
 2 28 31 35 40 49 51 58 68 77 61 78 95 82 99
 2 28 31 35 40 49 51 58 61 68 77 78 82 95 99 <- ab hier finden keine
 2 28 31 35 40 49 51 58 61 68 77 78 82 95 99    Vertauschungen mehr
 2 28 31 35 40 49 51 58 61 68 77 78 82 95 99    statt,
 2 28 31 35 40 49 51 58 61 68 77 78 82 95 99
 2 28 31 35 40 49 51 58 61 68 77 78 82 95 99
 2 28 31 35 40 49 51 58 61 68 77 78 82 95 99
```

*Abb. 8c: Bubblesort #1*

```
68 77 28 35 95 58 40 2 99 82 51 78 49 61 31
2 68 77 28 35 95 58 40 31 99 82 51 78 49 61
2 28 68 77 31 35 95 58 40 49 99 82 51 78 61
2 28 31 68 77 35 40 95 58 49 51 99 82 61 78
2 28 31 35 68 77 40 49 95 58 51 61 99 82 78
2 28 31 35 40 68 77 49 51 95 58 61 78 99 82
2 28 31 35 40 49 68 77 51 58 95 61 78 82 99
2 28 31 35 40 49 51 68 77 58 61 95 78 82 99
2 28 31 35 40 49 51 58 68 77 61 78 95 82 99
2 28 31 35 40 49 51 58 61 68 77 78 82 95 99
2 28 31 35 40 49 51 58 61 68 77 78 82 95 99 <- Erst in der ersten Zeile
                                               ohne Vertauschungen merkt
                                               Bubblesort #2, dass es
                                               fertig ist.
```

*Abb. 8d: Bubblesort #2*

Phantasie, die Array-Elemente als Blasen auffaßt, die gegen ihren Bestimmungsort im Array aufsteigen.
Die Prozedur finden Sie im Listing 8c realisiert. Der zugehörigen Abbildung können Sie leicht entnehmen, ob Sie die nötige Phantasie für die Akzeptanz der Namensgebung aufbringen.

## Bubblesort #2

Aus der Abbildung 8c läßt sich ebenfalls absehen, daß einige Sortierdurchgänge ohne Änderungen stattfinden (markierte Stelle). Eine leichte Änderung von **Bubblesort** schafft Abhilfe, indem ein Flag mit aufgenommen wird, welches sich die Vertauschung von Elementen merkt. Hat in einem Durchgang keine Vertauschung stattgefunden, so kann der Algorithmus abgebrochen werden. (Listing und Abbildung 8d)
Sicherlich ist es bei dem Bubblesort-Algorithmus und bei den beiden 'einfachen' Algorithmen sehr leicht möglich noch weitere Verbesserungen anzugeben, beispielsweise könnte man bei der direkten Auswahl auch gleichzeitig mit dem Minimum das Maximum bestimmen.
Wie die Sortierzeittestes aber ergeben werden, spielen alle diese Algorithmen nur eine untergeordnete Rolle, da sie bei großen Arrays viel zu langsam sind. Deshalb schenke ich mir weitergehende Verbesserungen.

## Shellsort

Anders ist das beim 1959 von **D.L.Shell** entwickelten Algorithmus **shellsort**. Shell ging davon aus, daß zunächst weiter auseinanderliegende Arrayelemente (Abstand k) verglichen und sortiert werden. Durch Reduzierung von k und abermaligen Austausch werden dadurch immer enger zusammenstehende Elemente verglichen. Letztendlich schließt **shellsort** mit k=1 und dem Vergleich benachbarter Elemente ab. Dies entspricht dann dem gewöhnlichen Sortieren.
Am besten macht man sich das an einem Beispiel, vorzugsweise unserem Standardbeispiel aus den vorhergehenden Abbildungen, klar. Das Resultat sehen Sie in Abbildung 8e. Die dabei vorgegebene Schrittfunktion lautet:

- *Beginne bei k:=max DIV 2*
- *Bei jedem weiteren Durchlauf verwende k:=k DIV 2*

Für unser Beispiel (max=15) führt dies zur Folge k=7,3,1. Zu Beginn, mit k=7, sind die jeweils sieben Positionen auseinanderliegenden Elemente

*(68,2); (77,99); (28,82); (35,51); (95,78); (58,49); (40,61) und (31,68)*

zu vergleichen, wobei jeweils das Minimum der zwei Elemente nach vorn rückt. Durch Fortführung des Verfahrens gelangt man schließlich zu k=1 und somit zum gewöhnlichen Sortieren. Dabei ist allerdings zu beachten, daß hier kaum noch Vertauschungen stattfinden, weil die vorhergehenden Stufen schon eine beträchtliche Vorarbeit geleistet haben.
Obwohl man, außer durch Zeitmessungen, sehr schlecht einsieht, daß **shellsort** gute Ergebnisse liefert - tatsächlich ist es bis heute noch nicht gelungen eine genaue mathematische Vorhersage über das Verhalten von **shellsort** zu gewinnen - ist von vorneherein klar, daß die Schrittweiten ein wichtiges Merkmal für die Güte des Sortierverfahrens darstellen.
Zur Beendung der Sortierarbeit ist es zwar lediglich notwendig, daß der Sortieralgorithmus mit der Schrittweite k=1 abgeschlossen wird, die vorhergehenden Schritte tragen allerdings sehr zur Sortiergeschwindigkeit bei.
Ohne weitere Begründung (die Mathematik läßt grüßen), gebe ich Ihnen deshalb zwei Schrittfunktionen an, die laut **Knuth** gute Ergebnisse liefern:

*A: Die Folge ..., 121, 40, 13, 4, 1*
*B: Die Folge ..., 31, 15, 7, 3, 1*

Wie Sie oben und im Listing 8e gesehen haben, wurde von mir keine dieser Schrittfunktionen gewählt, da das ein Schrittweitenarray erfordern würde, was, in notwendiger Allgemeinheit, recht umständlich zu implementieren ist. Auch die von mir gewählte Schrittfunktion lieferte, wie im Folgenden ausgeführt, recht brauchbare Ergebnisse.

## Testumgebung

Unsere heutige Testumgebung, Listing 8f, stellt uns drei mögliche Arten von Arrays zur Verfügung.

- Da wären zunächst zufällige Arrays, die mit Hilfe des entsprechenden Betriebssystemaufrufs generiert werden. Dabei wird das Ganze noch modulo **obergrenze** gerechnet, um bei den Demos möglichst kleine Schlüssel haben zu können (**obergrenze**=100) und bei den Tests ein möglichst breit gestreutes Spektrum (obergrenze=1000000) wählen zu können (Prozedur **make_random**).

```
        68 77 28 35 95 58 40 2 99 82 51 78 49 61 31
k=7 : 2 77 28 35 78 49 40 31 99 82 51 95 58 61 68
k=3 : 2 31 28 35 51 49 40 61 68 58 77 95 82 78 99
k=1 : 2 28 31 35 40 49 51 58 61 68 77 78 82 95 99
```

*Abb. 8e: Shellsort*

Zusätzlich können noch
- aufsteigend sortierte Arrays (Prozedur make_list)

und
- absteigend sortierte Arrays (Prozedur make_invlist)

generiert werden.

Maßgeblich sind dabei natürlich die zufällig erzeugten Arrays. Die Zeitwerte für die anderen Arrays werden lediglich benötigt, um das Verhalten der Algorithmen in Ausnahmesituationen (teil- oder ganzsortierte Arrays) zu verdeutlichen.

Um beim nächsten Mal nicht noch eine zusätzliche Testumgebung stricken zu müssen, sind auch schon die Sortieralgorithmen der nächsten Folge in **teste_sort** aufgenommen worden.

Zum reinen Test unserer heutigen Routinen wird es deshalb notwendig, daß Sie die Zeilen 17, 28-31, 104-107 und 116-119 ignorieren, sofern Sie sie nicht als Vorausschau auf Kommendes betrachten wollen.

Die Zeitmessung bereitete einige Schwierigkeit, da wir mit der Systemuhr nur in Zwei-Sekunden-Intervallen messen können. Heute ist dies noch nicht so problematisch, weil unsere Algorithmen, Ausnahme **shellsort**, noch recht schlecht sind, sich also lange Zeiten bereits bei einigen hundert Elementen ergeben können. Doch 'Lange Rede kurzer Sinn'. Jetzt wird's praktisch.

## Zeitmessung

Beginnen wir unsere Testreihe mit 500 Elementen und verdoppeln die Anzahl bis hin zu 8000 Elementen. Es ergeben sich für die fünf Sortieralgorithmen mit den drei möglichen Formen von Arrays (aufsteigend/ random/ absteigend) die Werte der Abbildung 8f.

Von eigentlichem Interesse sind dabei nur die Werte für die zufällig erzeugten Arrays, die wir nun etwas genauer betrachten wollen.

Beobachten können wir dabei zunächst, daß shellsort, sowieso schon weit und breit schnellster Algorithmus, mit wachsender Elementanzahl immer besser abschneidet und die anderen Verfahren, angeführt von den beiden bubblesort-Variationen, immer unakzeptablere Werte liefern.

Eine weitere Beobachtung ist die Art der Zeitentwicklung in Abhängigkeit von der Arraylänge. Verdoppelt man die Arraylänge, so ist bei allen Algorithmen, Ausnahme wiederum Shellsort, eine ungefähre Vervierfachung der Sortierzeiten ablesbar.

Es besteht also bei den vier Algorithmen eine quadratische Abhängigkeit der Sortierzeit von der Anzahl der Elemente. Diese Eigenschaft kann man durch die Formel:

*zeit=Konstante * Elemente*$^2$

charakterisieren. In diesem Fall spricht man von einem Algorithmus der Ordnung $O(n^2)$. Allgemein spricht man, wenn eine Begebenheit der Art

*zeit=Konstante * Term*

vorliegt, von einem Algorithmus der Ordnung O(Term).

Die quadratische Ordnung können Sie aus der Abbildung 8g auch sehr gut ablesen. Links dargestellt ist eine Ausgleichskurve durch unsere abgelesenen Werte, die, wie könnte es anders sein, Parabelform hat.

Um die ganze Sache noch zu überprüfen, kann man noch die zweiten Wurzeln aus den Sortierzeiten ziehen (rechter Abbildungsteil). Es müssen sich dann Geraden ergeben. Hier passen auch erstmals die sehr kurzen Sortierwerte von shellsort größenordnungsmäßig ins Bild. Dummerweise ergibt sich, was man trotz der sehr kurzen Zeiten gut erkennen kann, keine Gerade, sondern eine leicht abgeflachte Kurve. Dies liegt daran, daß shellsort bei geeigneter Schrittweitenfolge eine Ordnung unterhalb von $O(n^2)$, unbewiesenerweise etwa in der Nähe von $O(n^{1.2})$, besitzt.

| Anzahl | Array | Einfuegen | Auswahl | Bubblesort 1 | Bubblesort 2 | Shellsort |
|---|---|---|---|---|---|---|
| | ---> | - | 3.2 | 3.6 | - | - |
| 500 *) | Random | 2.4 | 3.2 | 5.6 | 6.0 | - |
| | <--- | 2.0 | 4.0 | 5.6 | 6.0 | - |
| | ---> | - | 12 | 14 | - | - |
| 1000 | Random | 10 | 12 | 22 | 24 | - |
| | <--- | 18 | 14 | 32 | 32 | - |
| | ---> | - | 50 | 52 | - | 2 |
| 2000 | Random | 42 | 52 | 92 | 92 | 2 |
| | <--- | 80 | 56 | 124 | 128 | 2 |
| | ---> | - | 210 | 214 | - | 2 |
| 4000 | Random | 168 | 210 | 370 | 378 | 6 |
| | <--- | 330 | 224 | 522 | 538 | 4 |
| | ---> | - | 840 | 856 | - | 4 |
| 8000 | Random | 666 | 840 | 1478 | 1512 | 14 |
| | <--- | 1316 | 898 | 2088 | 2156 | 6 |

*) : Mittelung von 5 Messungen
---> : Aufsteigend sortierter Array
Random : Zufallsarray mit Long_Integer-Werten von 0-1000000
<--- : Absteigend sortierter Array

*Abb. 8f: Sortiermessung (in Sekunden)*

## Vorausschau

In der nächsten Folge werden wir uns mit Sortieralgorithmen der Ordnung O(n*log(n)) befassen, deren günstigste Vertreter in der Praxis noch einmal eine ganze Ecke (Faktor 2-4) schneller als shellsort sind. Einige wohlklingende Namen in diesem Zusammenhang sind heapsort und quicksort.
Um eine iterative Variante von quicksort angeben zu können, werde ich zusätzlich darauf eingehen, wie man mit einem uns aus der ersten Folge wohlbekannten Hilfsmittel, den Stacks, sämtliche rekursiven Operationen iterativ darstellen kann. Ein Punkt, der die BASIC-Programmierer unter Ihnen (sind da noch welche ?) sicherlich sehr interessieren wird.

*Sig*

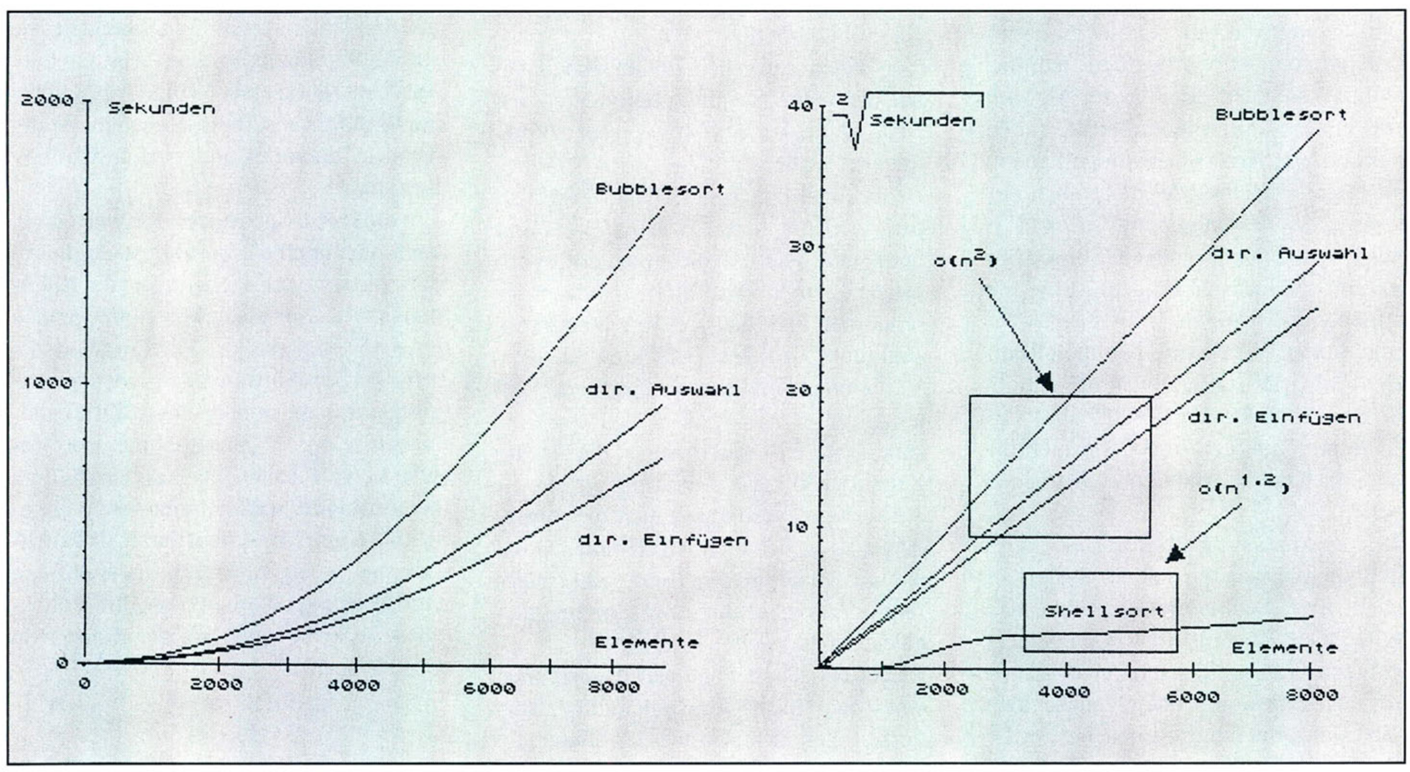



*Abb 8g: Zeitkennlinien (linear / quadratisch)*

```
 1:  { Listing 8a   : Sortieren mit direktem Einfuegen eines
 2:                   Elementes in eine schon sortierte Teilliste
 3:    programmiert : November '87
 4:    von          : Dirk Brockhaus
 5:    mit          : PASCAL+ (CCD)
 6:  }
 7:
 8:  PROCEDURE direktes_einfuegen;
 9:
10:    VAR i     ,
11:        j     : integer;
12:        help  : key_type;
13:
14:    BEGIN {direktes_einfuegen}
15:      WITH feld DO
16:        FOR i:=2 TO max DO
17:          BEGIN
18:            help:=a[i];
19:            j:=i-1;
20:            WHILE (j>0) AND (a[j]>help) DO
21:              BEGIN
22:                a[j+1]:=a[j];
23:                j:=j-1;
24:              END;
25:            a[j+1]:=help;
26:          END;
27:    END;  {direktes_einfuegen}
```

```
 1:  { Listing 8b   : Sortieren mit direkter Auswahl des
 2:                   Minimums einer unsortierten Teilliste.
 3:    programmiert : November '87
 4:    von          : Dirk Brockhaus
 5:    mit          : PASCAL+ (CCD)
 6:  }
 7:
 8:  PROCEDURE direkte_auswahl;
 9:
10:    VAR i     ,
11:        j     ,
12:        min   : integer;
13:        help  : key_type;
14:
15:    BEGIN {direkte_auswahl}
16:      WITH feld DO
17:        FOR i:=1 TO max-1 DO
18:          BEGIN
19:            min:=i;
20:            FOR j:=i TO max DO
21:              IF a[j]<a[min] THEN
22:                min:=j;
23:            help:=a[i];
24:            a[i]:=a[min];
25:            a[min]:=help;
26:          END;
27:    END;  {direkte_auswahl}
```

```
 1:  { Listing 8c   : Sortieren mit bubblesort.
 2:    programmiert : November '87
 3:    von          : Dirk Brockhaus
 4:    mit          : PASCAL+ (CCD)
 5:  }
 6:
 7:  PROCEDURE bubblesort_1;
 8:
 9:    VAR i     ,
10:        j     : integer;
11:        help  : key_type;
12:
13:    BEGIN {bubblesort_1}
14:      WITH feld DO
15:        FOR i:=2 TO max DO
16:          BEGIN
17:            FOR j:=max DOWNTO i DO
18:              IF a[j-1]>a[j] THEN
19:                BEGIN
20:                  help:=a[j-1];
21:                  a[j-1]:=a[j];
22:                  a[j]:=help;
23:                END;
24:          END;
25:    END;  {bubblesort_1}
```

```
 1:  { Listing 8d     : Sortieren mit bubblesort und Erkennung
 2:                     eines bereits sortierten ARRAY.
 3:    programmiert : November '87
 4:    von          : Dirk Brockhaus
 5:    mit          : PASCAL+ (CCD)
 6:  }
 7:
 8:  PROCEDURE bubblesort_2;
 9:
10:    VAR i    ,
11:        j    : integer;
12:        help : key_type;
13:        flag : boolean;
14:
15:    BEGIN {bubblesort_2}
16:      WITH feld DO
17:        BEGIN
18:          i:=2;
19:          flag:=false;
20:          WHILE (i<=max) AND NOT flag DO
21:            BEGIN
22:              flag:=true;
23:              FOR j:=max DOWNTO i DO
24:                IF a[j-1]>a[j] THEN
25:                  BEGIN
26:                    flag:=false;
27:                    help:=a[j-1];
28:                    a[j-1]:=a[j];
29:                    a[j]:=help;
30:                  END;
31:              i:=i+1;
32:            END;
33:        END;
34:    END;  {bubblesort_2}
```

```
 1:  { Listing 8e    : Sortieren mit shellsort und den
                       Schrittweiten
 2:                    k0 = max DIV 2 ; kn+1 = kn DIV 2.
 3:    programmiert : November '87
 4:    von          : Dirk Brockhaus
 5:    mit          : PASCAL+ (CCD)
 6:  }
 7:
 8:  PROCEDURE shellsort;
 9:
10:    VAR i    ,
11:        j    ,
12:        k    : integer;
13:        help : key_type;
14:
15:    BEGIN {shellsort}
16:      WITH feld DO
17:        BEGIN
18:          k:=max DIV 2;
19:          WHILE k>0 DO
20:            BEGIN
21:              FOR i:=k TO max DO
22:                BEGIN
23:                  j:=i-k;
24:                  WHILE (j>=0) AND  (a[j]>a[j+k]) DO
25:                    BEGIN
26:                      help:=a[j];
27:                      a[j]:=a[j+k];
28:                      a[j+k]:=help;
29:                      j:=j-k;
30:                    END;
31:                END;
32:              k:=k DIV 2;
33:            END;
34:        END;
35:    END;  {shellsort}
```

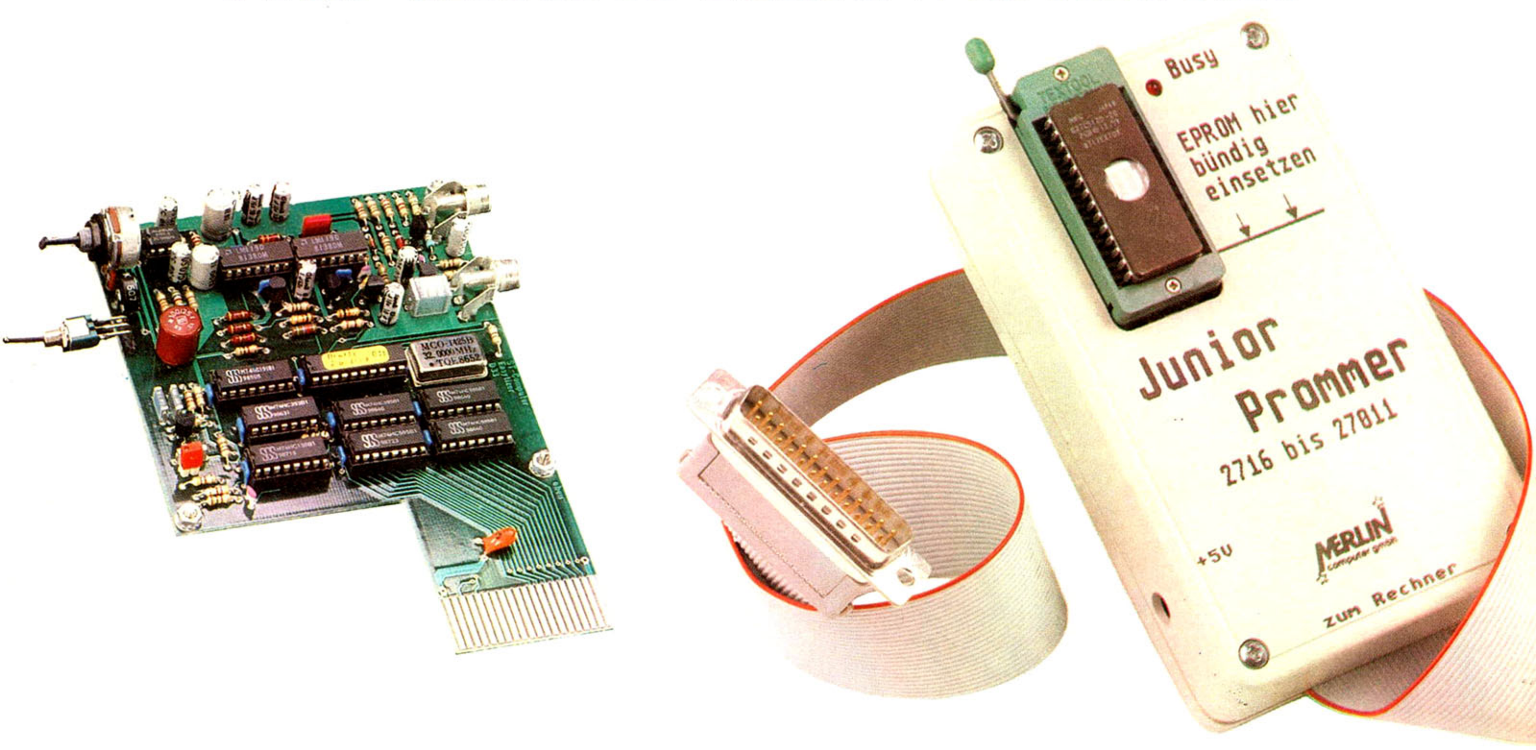
```
  1:  { Listing 8f    : Sortierzeitmessung einzelner Verfahren.
  2:    programmiert : November '87
  3:    von          : Dirk Brockhaus
  4:    mit          : Pascal+ (CCD)
  5:  }
  6:
  7:  PROGRAM teste_sort (input,output);
  8:
  9:  CONST obergrenze = 1000000;
 10:
 11:  TYPE  key_type   = long_integer;
 12:
 13:        field      = RECORD
 14:                       max : integer;
 15:                       a   : ARRAY [0..8000] OF key_type;
 16:                     END;
 17:
 18:        {$i stacktyp.pas}
 19:
 20:  VAR feld : field;
 21:
 22:  {$i stack.pas}
 23:
 24:  {$i direkt.pas}
 25:  {$i auswahl.pas}
 26:  {$i bubble_1.pas}
 27:  {$i bubble_2.pas}
 28:  {$i shell.pas}
 29:  {$i heap.pas}
 30:  {$i quick_1.pas}
 31:  {$i quick_2.pas}
 32:  {$i best.pas}
 33:
 34:  PROCEDURE teste_laufzeit;
 35:
 36:    VAR time : long_integer;
 37:        anz  ,
 38:        art  ,
 39:        sort ,
 40:        neu  : integer;
 41:
 42:    PROCEDURE make_random(anz : integer);
 43:
 44:      VAR i : integer;
 45:
 46:      FUNCTION random : long_integer;
 47:        XBIOS(17);
 48:
 49:      BEGIN {make_random}
 50:        WITH feld DO
 51:          BEGIN
 52:            max:=anz;
 53:            FOR i:=1 TO max DO
 54:              a[i]:=abs(random) MOD obergrenze;
 55:          END;
 56:      END;  {make_random}
 57:
 58:    PROCEDURE make_list(anz : integer);
 59:
 60:      VAR i : integer;
 61:
 62:      BEGIN {make_list}
 63:        WITH feld DO
 64:          BEGIN
 65:            max:=anz;
 66:              FOR i:=1 TO max DO
 67:                a[i]:=i;
 68:          END;
 69:      END;  {make_list}
 70:
 71:    PROCEDURE make_invlist(anz : integer);
 72:
 73:      VAR i : integer;
 74:
 75:      BEGIN {make_invlist}
 76:        WITH feld DO
 77:          BEGIN
 78:            max:=anz;
 79:              FOR i:=1 TO max DO
 80:                a[i]:=max-i+1;
 81:          END;
 82:      END;  {make_invlist}
 83:
 84:    BEGIN {teste_laufzeit}
 85:      writeln('Test der Laufzeit von neun
             unterschiedlichen');
 86:      writeln(' Sortierverfahren.');
 87:      writeln;
 88:      REPEAT
 89:        writeln('Anzahl Elemente (1..8000) ?');
 90:        readln(anz);
 91:        writeln('(1) Random');
 92:        writeln('(2) aufsteigend');
 93:        writeln('(3) absteigend');
 94:        write(' > '); readln(art);
 95:        CASE art OF
 96:          1 : make_random(anz);
 97:          2 : make_list(anz);
 98:          3 : make_invlist(anz);
 99:        END;
100:        writeln('(1) Direktes Einfuegen');
101:        writeln('(2) Direkte Auswahl');
102:        writeln('(3) Bubblesort #1');
103:        writeln('(4) Bubblesort #2');
104:        writeln('(5) Shellsort');
105:        writeln('(6) Heapsort');
106:        writeln('(7) Quicksort (rekursiv)');
107:        writeln('(8) Quicksort (iterativ)');
108:        writeln('(9) Bestsort (Quicksort & direktes
               Einfuegen)');
109:        write(' > '); readln(sort);
110:        time:=clock;
111:        CASE sort OF
112:          1 : direktes_einfuegen;
113:          2 : direkte_auswahl;
114:          3 : bubblesort_1;
115:          4 : bubblesort_2;
116:          5 : shellsort;
117:          6 : heapsort;
118:          7 : quicksort_1;
119:          8 : quicksort_2;
120:          9 : bestsort;
121:        END;
122:        writeln(' Sortierzeit: ',clock-time);
123:        write('Weitermachen (0/1) ? '); readln(neu);
124:      UNTIL neu=0;
125:    END;  {teste_laufzeit}
126:
127:  BEGIN {main}
128:    teste_laufzeit
129:  END.  {main}
```


EINSTEIN SYSTEMS
THOMAS KLUCKE ☎ 061 21/600172
6200 WIESBADEN
BUNSEN-STR. 6 e
ST
ATARI
NICK
ENDLICH DIE DATENVERWALTUNG FÜR AMATEUR-UND CB-FUNKER!
MIT STATISTISCHER AUSWERTUNG, QSL UND ANDERE DRUCKPRG.
FÜR 9/24-NADEL-DRUCKER, DM 89,-
LAUFZETTEL etc. HANDBUCH
DEMO GG. LEERDISK/FRANK RÜCKUMSCHLAG! PD
siehe DIESE AUSGABE DM 5/ST.
eins./doppelseitig (1-2, 3-4 usw.)
SUPERMATHEPRG. NATHAN ab August für AT lieferbar DM 99,-
Info anfordern! Testbericht Amiga Special-Sonderheft 1/88! Preise zzgl. Versandk. n. Menge + Warenart


Public-Domain Software
• Alle Programme werden auf erstklassigem Diskettenmaterial geliefert!
• PD-Software, siehe PD-Liste dieser Ausgabe.
Jede Einzeldiskette (SS oder DS formatiert)
nur: DM 5,99,–
Preis je 10er Block DS formatiert; 5 Disketten
nur: DM 29,99,–
Preis je 10er Block SS formatiert; 10 Disketten
nur: DM 49,99,–
Senden Sie Ihre Bestellung bitte an:
Active Software
Ostpreußenstr. 4
8400 Regensburg
[] Nachnahme: (Nur Inland zuzügl. DM 5,- Nachnahmegebühr).
[] Scheck über DM ............ liegt bei.
• Alle Preise inklusive Porto und Verpackung.


C.A.S.H.
T.i.M
Software
Cashflow
Ein Kassenbuch
Die zeitgemäße Form des Kassenbuchs
→ sofort: Kassenbestand
→ sofort: Negativbestand
→ sofort: Informationen
anfordern bei uns oder Ihrem Fachhändler

# Aus unserer Hardwareküche




## Easytizer - der Videodigitizer ohne Geheimnisse

Mit dem Easytizer können Sie beliebige Videosignale von einer Schwarzweiß- oder Farb-Kamera, Videorecorder oder direkt vom Fernsehgerät (mit Composite Video Ausgang) digitalisieren und somit auf dem Bildschirm Ihres ATARI ST sichtbar machen.
Der Easytizer wird am Modul-Port des ST angeschlossen. Die Auflösung beträgt 800x600 Bildpunkte, so daß in mittlerer Auflösung 640x200 Bildpunkte in vier Graustufen dargestellt werden können. In dieser Betriebsart werden 12,5 Bilder in der Sekunde wiedergegeben. Im hochaufgelösten Modus werden 640x400 Bildpunkte in Schwarzweiß wiedergegeben.

**Besondere Merkmale des Easytizers:**

- Software vollständig in Assembler
- Abspeichern der Bilder im DEGAS-Format
- Von STAD und Sympatic Paint ansteuerbar
- Ein beliebiger Bildausschnitt kann in ein mit der Maus wählbares Format vergrößert und verkleinert werden
- Spiegeln eines Bildes in horizontaler und vertikaler Ebene
- Animation mit beliebig vielen Bildern möglich, nur durch die Kapazität des Rechners begrenzt (beim Mega ST4 über 100 Bilder)
- Eingebauter Druckertreiber für die mittlere Auflösung für NEC P6/P7 und EPSON oder Kompatible
- Wahlweise automatische oder manuelle Helligkeitseinstellung
- Schnappschuß

**Lieferumfang:**

1) Fertiggerät
komplett aufgebaut und geprüft, inclusive Diskette mit der Easytizer Software und Bedienungsanleitung
2) Teilsatz
Für Bastler liefern wir einen Teilsatz bestehend aus:
- Doppelseitiger, elektronisch geprüfter Platine mit Lötstoplack und Bestückungskungsaufdruck sowie vergoldeten Anschlußkontakten
- fertig programmiertes GAL 16V8
- Quarzoszillatormodul 32 MHz
- Diskette und Bedienungsanleitung.

## Klein, kompakt und leistungsstark- der Junior Prommer

Der Junior Prommer programmiert alle gängigen EPROM-Typen, angefangen vom 2716 (2 KByte) bis zum modernen 27011 (1 MBit). Aber nicht nur EPROMs, sondern auch einige ROM- und EEPROM-Typen lassen sich lesen bzw. programmieren.
Zum Betrieb benötigt der Junior Prommer nur +5 Volt, die am Joystick-Port Ihres ATARI ST abgenommen werden, alle anderen Spannungen erzeugt die Elektronik des Junior Prommers. Die sehr komfortable Software, natürlich voll GEM unterstützt, erlaubt alle nur denkbaren Manipulationen.
Selbstverständlich läßt sich ein 16-Bit Word in ein High- und Low-Byte zerlegen. Fünf Programmieralgorithmen sorgen bei jedem EPROM-Typ für hohe Datensicherheit. Im eingebauten Hex/ASCII-Monitor läßt sich der Inhalt eines EPROMs blitzschnell durchsuchen oder auch ändern.

**Alles dabei!**
Bemerkenswert ist der Lieferumfang, so wird z.B. das Fertiggerät komplett aufgebaut und geprüft im Gehäuse mit allen Kabeln anschlußfertig geliefert. Auf der Diskette mit der Treibersoftware befinden sich noch RAM-Disk und ein Programm zum Erstellen von EPROM-Karten, ferner wird der Source-Code für Lese- bzw. Programmierroutinen mitgeliefert und last but not least ist im Bedienungshandbuch der Schaltplan abgedruckt.




Bestellcoupon Merlin Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Hiermit bestelle ich:

Name: ____________________
Vorname: ____________________
Straße: ____________________
Ort: ____________________
Unterschrift: ____________________

| | |
|---|---|
| ☐ Junior Prommer ( Fertiggerät wie beschrieben ) nur | DM 189,00 |
| ☐ Leerplatine und Software (o. Bauteile) | DM 49,00 |
| ☐ Leergehäuse (gebohrt und bedruckt) | DM 39,90 |
| ☐ ROM-Karte 128 KByte bietet maximal 4 EPROMS Platz (fertigbestückt o. EPROMs) | DM 58,00 |
| ☐ Easytizer (Fertiggerät) | DM 289,00 |
| ☐ Easytizer (Teilsatz wie oben beschrieben) | DM 129,00 |

Versandkosten: Inland DM 7,50
Ausland DM 10,00
Auslandbestellung **nur** gegen Vorauskasse. Nachnahme zuzgl. DM 3,50 Nachnahmegebühr.
☐ Vorauskasse
☐ Nachnahme

# Programmierpraxis



In dieser Folge unserer Programmierpraxis haben wir eine ganze Sammlung von kleinen Programm-Routinen parat. So ziemlich aus jeder gängigen Programmiersprache ist etwas dabei, so daß irgendetwas bestimmt Ihr Interesse findet. Allein schon der Schwarzweiß-Emulator an sich dürfte ein paar Freunde finden. Doch auch die kleine C-Programm-Routine Match!, die einen Teilstring in einem String findet (ähnlich dem BASIC-Befehl INSTR), ist recht nützlich.
Wer seine Adreßdaten aus BS-Handel gerne weiter in Adimens benutzen will, dem hilft hierbei ein kleines C-Programm, das die Daten entsprechend aufbereitet. Sicher gut zu gebrauchen, da Adimens ja wesentlich schneller ist als BS-Handel.
Um zu allen Routinen etwas zu sagen, sind es leider zu viele. Doch lassen Sie sich überraschen und schauen Sie selbst nach.

## INHALT

# SCHWARZ-WEIß-EMULATOR

**Mit diesem Programm läßt sich der Monochrom-Modus des ATARI ST auf einem Farbmonitor darstellen. Eine Vielzahl von Programmen (leider nicht alle) laufen mit diesem Schwarzweiß-Emulator. Natürlich erreicht man nicht die Qualität eines SM124, doch das Programm zeigt doch eine interessante Möglichkeit, wie man den ST manipulieren kann. Es ist allerdings notwendig, einige direkte Sprungadressen zu benutzen, so daß man einige Zeilen abändern muß, damit es auch auf dem Blitter-TOS läuft. Die Änderungen werden am Ende des Artikels angegeben.**

## Das Prinzip

Zuerst wird die TOS-Variable phystop ($42E) heruntergesetzt und dadurch Raum für einen zweiten Bildschirmspeicher und einige Routinen geschaffen, die nach einem Reset erhalten bleiben. Der Wert von phystop ändert sich bei einem Warmstart nicht.

Hinter phystop werden folgende Routinen abgelegt:

Eine veränderte vbl-Interrupt-Routine, bei der erstens der Monitor-Detect abgeschaltet ist (normalerweise führt der Computer ein Reset aus, wenn er merkt, daß im Schwarzweiß-Modus ein Farbmonitor angeschlossen ist oder andersherum) und die zweitens dafür sorgt, daß der Schwarzweiß-Bildschirmspeicher in den 2. Bildschirmspeicher hinter phystop übertragen und dabei auch gleich an die verschiedenen Modi angepaßt wird. Aus zeitkritischen Gründen wird dabei bei jedem vbl-Interrupt jeweils nur ein Viertel des Bildschirms übertragen, sonst würde der Computer extrem langsam werden.

Außerdem wird geprüft, ob sich vielleicht das Invers-Bit geändert hat. Die Farben paßt man dann entsprechend daran an. Auch eine Veränderung der Adresse des Schwarzweiß-Bildschirmspeichers wird erkannt und sofort gespeichert. Bei aktiven Floppies wird auch bei eingeschaltetem Emulator der Bildschirmspeicher nicht übertragen, da es sonst zu Fehlern kommen kann.

## Eine veränderte dump_vec-Routine

Über dump_vec springt der Computer jedesmal, wenn ALT/HELP gedrückt wird. Normalerweise erzeugt man damit eine Hardcopy auf dem Drucker. Hier dient dieser Vektor dazu, den Emulator ein- und auszuschalten (MIT Shift-Taste), bzw. zwischen den verschiedenen Modi hin- und herzuschalten (OHNE Shift-Taste). Beim Ausschalten des Emulator setzt diese Routine wieder die originale Bildschirmadresse.

## Ein veränderter Xbios-Aufruf

Folgende Xbios-Routinen werden verändert:

Xbios(4)=Getrez liefert immer den Wert 2 = High-Res zurück.

Xbios(2)=Physbase liefert bei eingeschaltetem Emulator die Adresse des Schwarzweiß-Bildschirmspeichers zurück.

Xbios(5)=Setscreen verändert bei eingeschaltetem Emulator nicht die wirkliche Bildschirmadresse (ab der der Videoprozessor abbildet), sondern die Adresse des Bildschirmspeichers, aus dem übertragen wird. Ein Ändern der Auflösung mit dieser Routine wird unterdrückt. Sonst verhält sich diese Routine wie gewöhnlich.

Bei allen anderen Xbios-Aufrufen werden die Original-Routinen angesprungen.

## Eine geänderte Keyboard-Interrupt-Routine

Diese sorgt nur dafür, daß während der Abarbeitung eines Keyboard-Interrupts der vbl-Interrupt eingeschaltet ist, da es sonst zu unschönen Erscheinungen kommen kann.

Außerdem werden hinter phystop noch einige Variablen, die der Emulator benötigt, gespeichert:

In alt_adr speichert er die Adresse des Schwarzweiß-Bildschirmspeichers, aus dem bei eingeschaltetem Emulator übertragen wird.

In neu_adr speichert er die Adresse des 2. Bildschirmspeichers hinter phystop, auf den bei eingeschaltetem Emulator der Videoprozessor gesetzt ist. Diese Adresse wird nur am Anfang gemerkt und dann nicht mehr verändert.

In aufl merkt er sich, ob er ein- oder ausgeschaltet ist. 1=An, 2=Aus.

In modus merkt er sich, in welchem der 4 Modi er sich befindet:

modus=0 : Jede 2. Zeile wird abgebildet (alle ungeraden Zeilen).

modus=1 : Jede 2. Zeile wird abgebildet (alle geraden Zeilen).

modus=2 : Der obere Teil des Bildschirms wird abgebildet.

modus=3 : Der untere Teil des Bildschirms wird abgebildet.

Nachdem das Programm die obigen 4 Routinen und die Variablen hinter phystop abgelegt hat, wird eine Routine erzeugt, die nach einem Reset automatisch aufgerufen wird. Wer Genaueres wissen will, sollte sich das ROM ab $fc0d20 ansehen. Eine Reset-Routine muß unterhalb von phystop an einer Adresse stehen, die ein Vielfaches von 512 ist. Das erste Langwort muß das "Magic" $12123456 sein, das zweite Langwort die Adresse des Magics enthalten und die Prüfsumme über 256 Worte = $5678 sein. Die Routine wird nach dem Laden des Boot-Sektors, aber noch vor dem Ausführen des Auto-Ordners, ausgeführt. In dieser Routine werden nur die Adressen der vier veränderten Aufrufe und die Auflösung = High gesetzt und dann die Bildschirmausgabe neu initialisiert.

Nach dem Laden des Programms prüft dieses zuerst, ob sich der Computer im Schwarzweiß-Modus befindet. Wenn ja,

wird das Programm sofort wieder verlassen und nichts geschieht. Im anderen Fall wird die Kurzanleitung ausgegeben und dann die Routine "supexec" im Supervisor-Modus angesprungen. Diese macht nun nichts anderes, als phystop heruntersetzen, die verschiedenen Routinen zu verschieben und die Reset-Routine zu erzeugen. Danach wird in einer Endlos-Schleife solange gewartet, bis der Resetknopf gedrückt wird.

## Änderungen für Blitter-TOS

Für all diejenigen, die mit dem Blitter-TOS arbeiten, müssen folgende Zeilen im Listing abgeändert werden:

| | | | |
|---|---|---|---|
| Zeile 88 | in: | btst | #1,$e61 |
| Zeile 180 | in: | jmp | $fc0748 |
| Zeile 182 | in: | jmp | $fc071a |
| Zeile 214 | in: | jmp | $fc07f2 |
| Zeile 235 | in: | jmp | $fc29ce |
| Zeile 252 | in: | jmp | $fca96e |

*Volker Ullrich*

```
  1:  ; Schwarzweiß Emulator
  2:  ;
  3:  ; (C) Volker Ullrich 1987
  4:  ; Postfach 1646
  5:  ; 7770 Überlingen
  6:  ;
  7:  ; bitte relozierbar anklicken
  8:
  9:  phystop     equ $42e
 10:  swv_vec     equ $46e
 11:  dump_vec    equ $502
 12:  sshiftmod   equ $44c
 13:
 14:   move    #4,-(a7)
 15:   trap    #14             ; xbios 4, Getrez
 16:   cmpi    #2,d0           ; Schon Highres ?
 17:   beq.s   term            ; ja = term
 18:   pea     message
 19:   move    #9,-(a7)
 20:   trap    #1              ; Gemdos 9, Print Line
 21:   pea     supexec
 22:   move.w  #38,-(a7)
 23:   trap    #14             ; xbios 38, Supexec
 24:  term:
 25:   clr     (a7)
 26:   trap    #1              ; gemdos 0, Term
 27:
 28:
 29:  supexec:
        ; Hier nachfolgende Routinen verschieben, Prüfsumme
 30:    ; erzeugen, phystop anpassen etc.
 31:   move.l  phystop,a0
 32:   suba.l  #$8000,a0  ; a0 = Adresse 2. Bildschirmbereich
 33:   move.l  a0,neu_adr
 34:   suba.l  #$200,a0   ; a0 = Adresse neues phystop
 35:   move.l  a0,phystop
 36:   move.l  a0,a1      ; phystop retten
 37:   suba.l  #$8200,a1  ; a1 = Adresse Reset-Routine
 38:   lea     -4(a1),a7  ; Stack heruntersetzen
 39:   move.l  #vbl_sprung,a2
 40:   move    #255,d0    ; Länge change_aufl + vbl2 : ca.
                            256 Worte
 41:  verschieben:
 42:   move.w  (a2)+,(a0)+  ; change_aufl hinter phystop
                              kopieren
 43:   dbra    d0,verschieben
 44:   move.l  a1,a0        ; a0 = Anfang Reset-Routine
 45:   move    #127,d0      ; 128 Langworte
 46:  loeschen:
 47:   clr.l   (a0)+        ; Platz für Reset-Routine löschen
 48:   dbra    d0,loeschen
 49:   move.l  a1,a0        ; a0 = Anfang Reset-Routine
 50:   move.l  #reset_resident,a2
 51:   move    #39,d0       ; Länge: ca. 40 Worte
 52:  verschieben2:
 53:   move.w  (a2)+,(a0)+  ; Reset-Routine verschieben
 54:   dbra    d0,verschieben2
 55:   move.l  a1,4(a1)     ; Adresse der Routine speichern
 56:   move    #255,d0      ; 256 Worte für Prüfsumme
 57:   clr     d1           ; Prüfsumme löschen
 58:  pruefsumme:
 59:   add     (a1)+,d1     ; Prüfsumme erzeugen
 60:   dbra    d0,pruefsumme
 61:   move    #$5678,d0
 62:   sub     d1,d0
 63:   move    d0,-(a1)     ; Prüfsumme korrigieren
 64:  endlos:
 65:   bra.s   endlos       ; Auf Reset warten
 66:
 67:
 68:
 69:  ; Die folgenden Routinen werden hinter phystop
        gespeichert :
 70:
 71:  vbl_sprung:
 72:   bra     vbl2         ; geänderter vbl Interrupt Aufruf
```

```
 73:   bra     xbios2        ; geänderter xbios Aufruf
 74:   bra     keyb_int      ; geänderter mfp 68901 Level 6-
                               Interrupt
 75:   bra.s   change_aufl   ; geänderter dump_vec
 76:
 77:  alt_adr:
 78:   dc.l    0
 79:  neu_adr:
 80:   dc.l    0
 81:  aufl:
 82:   dc.b    2             ; default: High=Emulator aus
 83:  modus:
 84:   dc.b    0             ; Modus : Jede 2. Zeile a)
 85:
 86:  change_aufl:              ; Aufruf dieser Routine mit
                                  Alt/Help
 87:   lea     alt_adr(pc),a0  ; Anfang Variablen in a0
 88:   btst    #1,$e1b         ; linke Shifttaste?
 89:   bne.s   shift           ; ja
 90:   addq.b  #1,9(a0)  ; wenn nicht gedrückt: modus erhöhen
 91:   andi.b  #3,9(a0)  ; falls 4 erreicht, wieder modus=0
 92:   rts
 93:  shift:
 94:   eori.b  #3,8(a0)    ; Hier Auflösung Umschalten
 95:   cmpi.b  #1,8(a0)    ; Neue Auflösung = Mid ?
 96:   bne.s   aufl_high   ; Nein
 97:   moveq.l #0,d0       ; Hier wenn neue Auflösung Mid
 98:   move.b  $ff8201,d0
 99:   lsl.w   #8,d0
100:   move.b  $ff8203,d0
101:   lsl.l   #8,d0       ; physikalische Videoadr. in d0
102:   move.l  d0,(a0)     ; in alt_adr merken
103:   move.l  4(a0),-(a7) ; Neue Videoadresse auf Stack
104:  video_setzen:
105:   move.b  1(a7),$ff8201   ; und setzen
106:   move.b  2(a7),$ff8203
107:   clr.l   (a7)+           ; Stack löschen
108:   rts
109:  aufl_high:
110:   move.l  (a0),-(a7)      ; Alte Videoadresse auf Stack
111:   bra.s   video_setzen    ; und setzen
112:
113:
114:  vbl2:       ; veränderte vbl-Routine ohne Monitor-Detect
115:   movem.l d0-d7/a0-a6,-(a7) ; Register retten
116:   lea     alt_adr(pc),a0    ; Anfang Variablen in a0
117:   move.b  8(a0),d0          ; Auflösung in d0
118:   move.b  d0,$ff8260        ; und setzen
119:   cmpi.b  #1,d0             ; Mid Auflösung ?
120:   bne     nix               ; Nein : nichts verschieben
121:   move.l  #$ffff,d0         ; Hier wenn Auflösung Mid
122:   btst    #0,$ff8241        ; Bit Null Farbpalettenregister
                                    0 gesetzt ?
123:   bne.s   invers            ; ja
124:   swap    d0                ; sonst : Maske vertauschen
125:  invers:
126:   move    d0,$ff8240        ; schwarze und weiße Farbe
                                    setzen
127:   swap    d0
128:   move    d0,$ff8246
129:   moveq.l #0,d0            ; aktuelle Videoadresse ausrechnen
130:   move.b  $ff8201,d0
131:   lsl.w   #8,d0
132:   move.b  $ff8203,d0
133:   lsl.l   #8,d0
134:   cmp.l   4(a0),d0      ; muß bei Mid = neu_adr sein
135:   beq.s   gleich
136:   move.l  d0,(a0)       ; sonst: alt_adr neu setzen
137:   move.l  4(a0),-(a7)   ; neu_adr auf Stack
138:   move.b  1(a7),$ff8201 ; und setzen
139:   move.b  2(a7),$ff8203
140:   clr.l   (a7)+         ; Stack löschen
141:  gleich:
142:   tst.w   $43e          ; flock, Floppies aktiv ?
143:   bne.s   nix           ; Ja : nichts verschieben
144:   move.w  $468,d0       ; vbl-Zähler
```

```
145:  andi.w  #3,d0
146:  mulu    #8000,d0      ; Jeden vbl ein Viertel übertragen
147:  move.l  4(a0),a1      ; Neue Videoadresse in a1
148:  move.l  (a0),a2       ; Alte Videoadresse in a2
149:  move.b  9(a0),d1      ; Modus in d1
150:  bne.s   modus1
151:  lea     80(a2),a2     ; 2. Zeilensatz
152: modus1:
153:  cmpi.b  #3,d1
154:  bne.s   modus2
155:  adda.l  #16000,a2     ; untere Bildschirmhälfte
156: modus2:
157:  adda.l  d0,a1
158:  moveq.l #80,d2        ; Jede 2. Zeile verschlucken
159:  btst    #1,d1
160:  beq.s   modus3
161:  clr.l   d2            ; Modus 2,3: Jede 2. Zeile nicht
                              verschlucken
162:  asr.l   #1,d0         ; d0/2
163: modus3:
164:  adda.l  d0,a2
165:  move    #49,d1        ; 50 Zeilen
166: schleife1:
167:  move    #39,d0        ; 80 Bytes je Zeile
168: schleife2:
169:  move    (a2),(a1)+
170:  move    (a2)+,(a1)+
171:  dbra    d0,schleife2
172:  adda.l  d2,a2           ; Jede 2. Zeile verschlucken
173:  dbra    d1,schleife1
174: nix:
175:  addq.l  #1,$466         ; wie Original
176:  subq.w  #1,$452         ; wie Original
177:  bmi.s   gesperrt        ; wie Original
178:  addq.l  #1,$462         ;     "
179:  sub.l   a5,a5           ;     "
180:  jmp     $fc069e         ; und Sprung in Original Routine
                          (aber hinter der kritischen Stelle)
181: gesperrt:
182:  jmp     $fc071a         ; vbl gesperrt: Sprung an Ende
                                der Routine
183:
184: ; Obige zwei Sprungadressen gelten nur für ROM-TOS !!
185:
186:
187: xbios2:
188:  lea     6(a7),a0        ; Adresse Funktionsnummer in a0
189:  btst    #5,(a7)         ; Aufruf aus Supervisor ?
190:  bne.s   supervisor      ; ja
191:  move.l  usp,a0          ; Adresse Funktionsnummer in a0
192: supervisor:
193:  lea     alt_adr(pc),a1  ; Adresse Variablen in a1
194:  cmpi    #4,(a0)         ; = 4 ?  (Getrez)
195:  bne.s   weiter          ; nein
196:  moveq.l #2,d0           ; Getrez = 2
197:  rte
198: weiter:
199:  cmpi    #2,(a0)         ; = 2 ?  (Physbase)
200:  bne.s   weiter2         ; nein
201:  move.l  alt_adr(pc),d0  ; Physbase = alt_adr
202:  cmpi.b  #1,8(a1)        ; Emulator an ?
203:  beq.s   physb_aus       ; Ja
204:  moveq.l #0,d0           ; sonst: aktuelle Videoadresse
                                ausrechnen
205:  move.b  $ff8201,d0
206:  lsl.w   #8,d0
207:  move.b  $ff8203,d0
208:  lsl.l   #8,d0
209: physb_aus:
210:  rte
211: weiter2:
212:  cmpi    #5,(a0)       ; = 5 ?  (Setscreen)
213:  beq.s   setscreen
214:  jmp     $fc0748       ; Original xbios, nur für ROM-TOS
215: setscreen:
216:  tst.l   2(a0)
217:  bmi.s   setscreen2
218:  move.l  2(a0),$44e    ; _v_bs_ad setzen
219: setscreen2:
220:  tst.l   6(a0)
221:  bmi.s   setscr_aus
222:  move.l  6(a0),(a1)    ; physbase in alt_adr speichern
223:  cmpi.b  #1,8(a1)      ; Emulator an ?
224:  beq.s   setscr_aus    ; Ja
225:  move.l  6(a0),-(a7)   ; Sonst: auch Videoadresse setzen
226:  move.b  1(a7),$ff8201
227:  move.b  2(a7),$ff8203
228:  clr.l   (a7)+         ; Stack löschen
229: setscr_aus:           ; Auflösung ändern wird unterdrückt
230:  rte
231:
232:
233: keyb_int:
234:  andi.b  #$fb,(a7)      ; Interruptmaske 0-3, vbl enable
235:  jmp     $fc281c            ; Sprung in Original Routine
236:
237:                      ; obige Adresse gilt nur für ROM-TOS !
238:
239:
240: reset_resident:
  ; Wird an 512 Grenze kopiert und nach Reset ausgeführt
241:  dc.l $12123456     ; Magic
242:  dc.l 0             ; Hier kommt Adresse des Magic hinein
243:  move.l phystop,a0
244:  move.l a0,$70      ; Level 4 Interrupt : Auf vbl2
245:  lea    4(a0),a0    ; a0 += 4
246:  move.l a0,$b8      ; Trap #14 : Auf xbios2
247:  lea    4(a0),a0
248:  move.l a0,$118     ; mfp 68901, Level 6 auf keyb_int
249:  lea    4(a0),a0
250:  move.l a0,$502     ; dumpvec auf change_aufl
251:  move.b #2,sshiftmod  ; Auflösung setzen
252:  jmp    $fca7c4     ; Bildschirmausgabe initialisieren
                         + rts
253:
254:                     ; Adresse gilt nur für ROM-TOS !
255:
256:
257:  data
258: message:
259:  dc.b 27,"v",27,"E"    ; Überlauf ein & Cls
260:  dc.b "Schwarz-Weiß Emulator © 1987 Volker Ullrich",
                                                      13,10
261:  dc.b "Dieses Programm simuliert auf einem Farbmonitor
              einen Schwarzweiß Monitor.",13,10,13,10
262:  dc.b "Bedienung:",13,10,13,10
263:  dc.b "Shift/Alternate/Help --> Emulator An/Aus",13,10
264:  dc.b "Alternate/Help --> Modus umschalten",13,10
265:  dc.b "Da ein Farbmonitor nur die Hälfte der
         Zeilenanzahl eines Schwarz-Weiß Monitors",13,10
266:  dc.b "darstellen kann (640x200 anstatt
       640x400 Punkte), wird mit Alt/Help zwischen",13,10
267:  dc.b "4 Möglichkeiten umgeschaltet:",13,10
268:  dc.b "1.) Nur jede 2. Zeile wird abgebildet
                          (1., 3., 5., 7. usw.)",13,10
269:  dc.b "2.) Wie oben, jedoch 2., 4., 6. usw. Zeile",
                                                      13,10
270:  dc.b "3.) Nur die obere Hälfte des Bildschirms
                                  wird abgebildet",13,10
271:  dc.b "4.) Nur die untere Hälfte des Bildschirms
                            wird abgebildet",13,10,13,10
272:  dc.b "Funktioniert nur mit ROM-TOS !",13,10
273:  dc.b "Kann auch in den Auto-Ordner kopiert werden.",
                                                13,10,13,10
274:  dc.b "Jetzt bitte eine Diskette einlegen und den
                   Resetknopf drücken. Der Computer geht"
275:  dc.b "dann automatisch nach kurzer Zeit in den
          Hochauflösenden Modus. Nach Beendigung",13,10
276:  dc.b "des Ladevorgangs Shift/Alt/Help drücken, um den
                    Emulator einzuschalten.",13,10,13,10,0
277:
278:
279:  end
```



*ENDE*

# UNDO-Routinen in GFA-BASIC

**Den Anwendern von Mal- und Zeichenprogrammen ist die UNDO-Funktion, die mit der gleichnamigen Taste des Atari ST ausgelöst wird, wohlbekannt: Diese Taste ermöglicht es, die zuletzt ausgeführte graphische Funktion auf dem Bildschirm rückgängig zu machen. Damit können Bilder gerettet werden, die sonst, z.B. durch ein falsches Füllmuster oder eine falsch positionierte Linie, verloren wären.**

In diesem Artikel werden die Grundlagen der Programmierung einer universellen UNDO-Routine in GFA-BASIC dargestellt. Bei der Entwicklung dieser Routine bildete die Programmierung einer mausgesteuerten Notenedition den Ausgangspunkt. Die Positionierung von Noten auf dem Bildschirm mittels Maus ist an sich eine komfortable Vorgehensweise. Aufgrund der Vielfalt der graphischen Symbole sowie der äußerst engen räumlichen Abstände sind Eingabefehler jedoch kaum zu vermeiden. Derartige Fehler können mit der UNDO-Routine problemlos korrigiert werden. Die vorgestellte Routine ist dabei ohne weiteres auf andere Programme, innerhalb derer graphische Symbole auf engem Raum positioniert werden, übertragbar.
Da das folgende Programmlisting lediglich Beispielcharakter haben soll, wird die Funktionsweise der UNDO-Routine hier im Rahmen einer einfachen Graphik-Prozedur demonstriert. In dieser Prozedur können Noten an beliebiger Stelle in einem Notensystem positioniert werden.

## Prinzip

Das Prinzip der UNDO-Funktion besteht darin, jeweils vor der nächsten Ausführung eines graphischen Befehls den Bildschirm komplett zu retten, d.h. zu speichern. Im Falle eines Eingabefehlers wird dann das neue, fehlerhafte Bild einfach durch das alte, fehlerfreie Bild ersetzt.

## SGET und SPUT

Da in der UNDO-Routine nicht ein Bildschirmausschnitt, sondern immer der komplette Bildschirm - einschließlich Menüleiste - gespeichert wird, bieten sich für diesen Zweck die sehr schnellen Befehle SGET und SPUT des GFA-BASIC an.
Der mit SGET in einer Stringvariablen gespeicherte Bildschirminhalt wird mit dem SPUT-Befehl in den Bildschirmspeicher transportiert. Sein alter Inhalt wird dabei überschrieben, so daß der fehlerhafte Bildschirm innerhalb der UNDO-Routine nicht mit CLS gelöscht zu werden braucht - SPUT erledigt diese Aufgabe automatisch.

## Realisierung

Es sind drei Problemstellungen zu lösen:

1. Der SGET-Befehl darf nur unmittelbar vor einem Graphik-Befehl ausgeführt werden, da andernfalls eine ordnungsgemäße UNDO-Funktion nicht gewährleistet ist.
2. In Programmmen mit Desktop-Menü müssen UNDO-Tastenabfrage und On Menü-Ereignisüberwachung voneinander getrennt werden, weil sonst die UNDO-Funktion durch zeitliche Verzögerungen beeinträchtigt wird.
3. Der Graphikbefehl soll nur einmal pro Mausklick ausgeführt werden.

Es wird davon ausgegangen, daß das Listing in einem Programm mit Desktop-Menü und GEM-Fenstern verwendet wird.

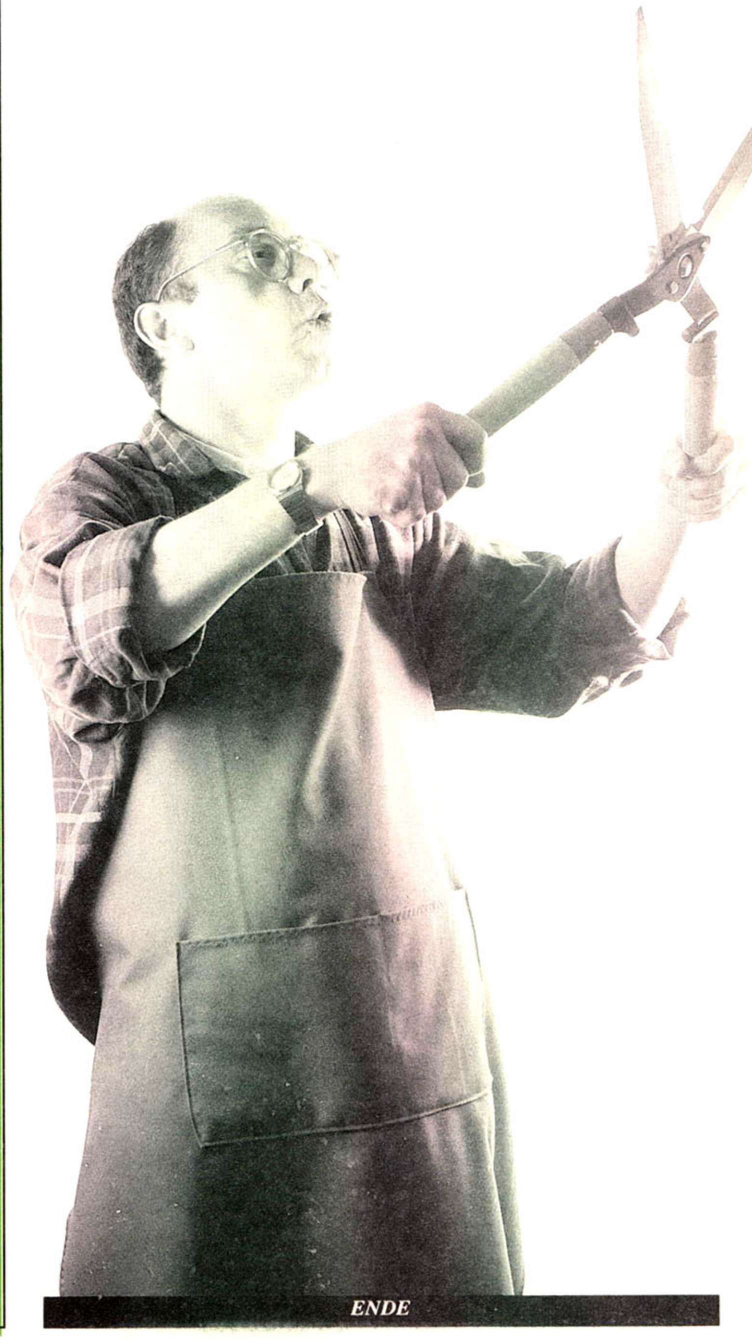
Für den Fall, daß die UNDO-Taste versehentlich vor dem ersten Mausklick betätigt wird, muß zunächst die Variable >Bild$< durch >Sget Bild$< gefüllt werden. Träfe nämlich der Befehl >Sput Bild$< auf eine noch nicht definierte Variable, wäre eine 2-Bomben-Fehlermeldung die unangenehme Folge.
In einer >Do-Loop<- Hauptschleife befindet sich die eigentliche UNDO-Schleife. Bis zur Betätigung einer Maustaste findet in dieser Schleife die Abfrage der UNDO-Taste mit >If Inkey$ = Chr$(0) + Chr$(97) < statt.
Wird die Prozedur in einem Programm mit Desktop-Menü und GEM-Fenster eingesetzt, ist folgendes zu berücksichtigen: Wenn in der UNDO-Schleife Ereignisüberwachung und Tastaturabfrage nicht getrennt sind, äußert sich dies in der Weise, daß die UNDO-Taste mit zeitlicher Verzögerung "anspricht". Deshalb wird durch eine Bedingungsabfrage sichergestellt, daß nur dann, wenn der Mauszeiger den Bereich der Menü-Titelzeile betritt, die Ereignisüberwachung mit >On menu< eingeschaltet wird. Hierbei ist noch zu bemerken, daß für die Y-Mauskoordinate oberhalb des Arbeitsbereichs - also im Bereich von Infozeile, Titelzeile und Menütitelzeile - Werte von 65498 bis 65534 zurüchgegeben werden. Aufgrund dieses Kuriosums ist der Graphikfenster-Bereich durch Y%-Werte >30 und <399 eingegrenzt.
Erfolgt jetzt im Programm ein Mausklick mit der linken oder rechten Taste, wird die UNDO-Schleife verlassen. Für graphische Funktionen ist es im Hinblick auf die Ökonomie des Listings sinnvoll, die horizontale und vertikale Mauskoordinate sowie den Status der Maustasten mit >Mouse< den Variablen X%,Y% und K% zuzuordnen. Diese Variablen werden in die folgenden Graphikbefehle >Pellipse< und >Line< eingebunden. Unmittelbar bevor nun mit diesen Befehlen eine einzelne Note auf dem Bildschirm plaziert wird, findet mit >Sget Bild$< die Speicherung des noch nicht veränderten Bildschirms statt. Die Richtung des Notenhals wird dabei mit der linken Maustaste (Hals nach oben) bzw. der rechte Taste(Hals nach unten) bestimmt.
Danach verbleibt das Programm bei gedrückter Maustaste in einer Warteschleife. Damit ist gewährleistet, daß pro Mausklick die Graphikbefehle >Pellipse< und >Line< nur einmal ausgeführt werden, d.h. pro Mausklick genau eine Note gezeichnet wird. Nach dem Loslas-

sen der Maustaste gelangt das Programm aus der Do-Loop-Hauptschleife wieder zurück in die UNDO-Schleife. Jetzt kann mit Betätigung der UNDO-Taste die letzte Bildschirmveränderung zurückgenommen, bzw. mit erneutem Mausklick die nächste Note positioniert werden.

Die UNDO-Routine kann in der gezeigten Form unverändert in weitere Prozeduren eines Programms übernommen werden. Sofern sich diese graphischen Prozeduren auf den gleichen Bildschirm beziehen, kann der SGET-Befehl auch mit der gleichen Variablen - im Programmbeispiel >Bild$< - verknüpft werden. Auf diese Weise wird der Speicherbedarf der UNDO-Funktion im gesamten Programm auf 32 kByte begrenzt.

*Hans-H. Ackermann*

```
 1:  Dim M$(17)                 !Menü erstellen
 2:  Restore Mmenue
 3:  For Z%=0 To 16
 4:    Read M$(Z%)
 5:  Next Z%
 6:  Mmenue:
 7:  Data DESK  ,  UNDODEMO.--------------------
 8:  Data 1,2,3,4,5,6,""
 9:  Data NOTEN  ,  Zeichnen,----------------,  Quit,"","",""
10:  Menu M$()
11:  Titlew 1," UNDODEMO "  !GEM-Fenster
12:  Fullw 1                  !öffnen
13:  '
14:  Gosub Menu
15:  Procedure Menu
16:    Do                            !Auf Ereignis
17:      On Menu                     !warten
18:      On Menu  Gosub Auswahl      !und verzweigen
19:    Loop
20:  Return
21:  '
22:  Procedure Auswahl               !Auswahl treffen
23:    Do
24:      If Menu(0)=1
25:        Menu Off
26:        Gosub Copyright
27:      Endif
28:      If Menu(0)=11
29:        Menu Off
30:        Gosub Noten               !Noten zeichnen
31:      Endif
32:      If Menu(0)=13
33:        Menu Off
34:        End                       !Ende
35:      Endif
36:    Loop
37:  Return
38:  '
39:  ' **** UNDO-Demo ****
40:  '
41:  Procedure Noten
42:    Gosub Noten_system
43:    Sget Bild$
44:    Do
45:      While Mousek=0              !UNDO-Schleife
46:        If Inkey$=Chr$(0)+Chr$(97)  !Wenn UNDO-Taste
                                       gedrückt
47:          Sput Bild$                !Bildschirmspeicher
                                       erneuern
48:        Endif
49:        If Mousey<30 Or Mousey>399  !Maus in Titelzeile?
50:          On Menu
51:        Endif
52:      Wend
53:      Mouse X%,Y%,K%
54:      If K%=1 And Y%>30 And Y%<399  !Linke Maustaste
                                       gedrückt?
55:        Sget Bild$                  !Altes Bild retten
56:        Pellipse X%,Y%,5,3          !Notenkopf und
57:        Line X%+5,Y%,X%+5,Y%-15     !Notenhals (nach oben)
                                       zeichnen
58:      Endif
59:      If K%=2 And Y%>30 And Y%<399  !Rechte Maustaste
                                       gedrückt?
60:        Sget Bild$                  !Altes Bild retten
61:        Pellipse X%,Y%,5,3          !Notenkopf und
62:        Line X%-5,Y%,X%-5,Y%+15     !-hals (nach unten)
                                       zeichnen
63:      Endif
64:      While Mousek<>0               !Warteschleife
65:      Wend
66:    Loop
67:  Return
68:  '
69:  '
70:  Procedure Noten_system            !Zeichnet
                                       Notenlinien
71:    Cls
72:    Local Z%
73:    Y0%=100
74:    For Z%=0 To 4
75:      Line 50,Y0%+6*Z%,550,Y0%+6*Z%
76:    Next Z%
77:  Return
78:  '
79:  Procedure Copyright
80:    Cls
81:    Box 190,70,420,270
82:    Deftext ,,,26
83:    Text 200,130,"UNDO-Routinen"
84:    Deftext ,,,13
85:    Text 260,170,"in GFA-Basic"
86:    Deftext ,,,6
87:    Text 220,210,"von Hans-H.Ackermann"
88:    Gosub Menu
89:  Return
```



**ENDE**

# NACHBRENNER FÜR SAVE/ LOAD VON FELDERN IN GFA-BASIC

**Entstanden sind diese Routinen zum Speichern und Laden von Feldern, weil sich mein sonst so schneller Computer beim Speichern des Feldes Feld%(249,249) wohl in eine Schildkröte verwandelt haben mußte.**

Für Schreiben über PRINT# und Lesen über INPUT# brauchte er sage und schreibe:

speichern : 586,715 s = 9,47 min
laden : 514,665 s = 8,35 min

Was für meinen Geschmack deutlich zu lange war.

Doch da sollte sich etwas ändern. Beschäftigt man sich mit der Speicherverwaltung des ST, so erkennt man, daß die Felder sozusagen am Stück im Speicher vorliegen. Daraufhin meine Idee, das ganze Feld als Ganzes abzuspeichern und genauso wieder zu laden.

Zum Abspeichern benötigt man also die Startadresse (Start%) im Speicher und die Länge des Feldes (Lange%). Beides ist nicht all zu schwierig.

Die Startadresse läßt sich mit der Funktion Arrptr() feststellen, danach folgen nur noch die Tiefen der einzelnen Dimensionen, und direkt danach fängt das Feld an.

Die Länge ergibt sich, indem man die Tiefen der einzelnen Dimensionen und die Länge einer einzelnen Variablen miteinander multipliziert.

Das Laden vereinfacht sich durch das ganze noch mehr. Einfach mit Varptr(Feld%(0,0)) den Pointer auf die Startadresse des Feldes holen und dann das vorher gespeicherte Feld wieder direkt an die richtige Stelle im Speicher laden.

Messungen auf einer leeren Diskette ergaben für das obige Feld

speichern : 62,4 s ==> 9,4*so schnell
laden : 27,2 s ==> 18,9*so schnell

Ganz beachtlich, oder etwa nicht?

Ein ganz netter Nebeneffekt trat auch noch auf, denn das Ganze braucht auf der Diskette auch nur noch 250kB statt der satten 410kB mit denen das File vorher zu Buche schlug. Das entspricht einer Reduktion auf 60%.

## Anpassung

Was muß ich jetzt jeweils für meine eigenen Bedürfnisse anpassen? Die Routine erkennt fast alles von selbst, man muß nur die Länge der jeweiligen Variablen anpassen. In der Routine ist dies schon vorgesehen. Man muß nur die richtige Zeile verwenden. Und sonst einfach den Variablennamen FELD% durch den eigenen Feldnamen ersetzen.

Und sonst: Ein Problem gab es bei der Sache doch noch, denn mit String Feldern funktionert das ganze nämlich nicht.Der Grund liegt darin, daß die Strings nicht mit einer festgegebenen Länge abgelegt werden , sondern abhängig von der Länge der Strings.

*Rolf Köhling*

```
 1:  Dim Feld%(249,249)
 2:  '
 3:  For I%=0 To 249
 4:    For J%=0 To 249
 5:      Feld%(J%,I%)=J%*100+I%
 6:    Next J%
 7:  Next I%
 8:  '
 9:  Ti1=Timer
10+  Gosub Speichern
11:  Ti2=Timer
12:  '
13:  Ti3=Timer
14:  Gosub Laden
15:  Ti4=Timer
16:  '
17:  Print "             ZEIT :    NEU          ALT"
18:  Print "Speichern         : ";(Ti2-Ti1)/200;" s       586.
     715 s"
19:  Print "Laden             : ";(Ti4-Ti3)/200;" s       514.
     665 s"
20:  '
21:  '
22:  Procedure Speichern
23:    Local Byte%,Feldadr%,Anzahl%,Start%,Lange%,I%
24:    '
25:    ' Byte%=2   ! für X!
26:    Byte%=4     ! für X%
27:    ' Byte%=6   ! für X
28:    '
29:    Feldadr%=Lpeek(Arrptr(Feld%()))
30:    Anzahl%=Dpeek(Arrptr(Feld%())+4)
31:    Start%=Feldadr%+4*Anzahl%
32:    '
33:    Lange%=1
34:    For I%=1 To Anzahl%
35:      Mul Lange%,Lpeek(Feldadr%+4*Anzahl%-4*I%)
36:    Next I%
37:    Mul Lange%,Byte%
38:    '
39:    Bsave "DATEN.DAT",Start%,Lange%
40:    '
41:  Return
42:  '
43:  Procedure Laden
44:    '
45:    Bload "DATEN.DAT",Varptr(Feld%(0,0))
46:    '
47:  Return
```

ENDE

# MATCH!

**In der Computerei spielt das Auffinden von Teilstrings innerhalb eines Strings eine zentrale Rolle. Ein kleines C-Prögrämmchen erledigt diesen Job. Daß es dabei auch die Verwendung von Wildcards erlaubt, versteht sich von selbst...**

Fast jede Programmiersprache hält irgendeinen Befehl bereit, mit dessen Hilfe man Substrings finden kann. INSTR, POS, SUBSEG sind einige der bekannteren Vertreter; UNIX stellt für derlei Abfragen die Prozedur fgrep zur Verfügung. In C dagegen gibt es standardmäßig nur die beiden Funktionen index() und rindex(), die auf das erste bzw. letzte Vorkommen eines Zeichens innerhalb einer Zeichenkette prüfen. Mager.

## Wilde Karten - neu gemischt

Die Funktion match() ist nicht nur in der Lage, Substrings ausfindig zu machen, sondern sie erlaubt auch die Verwendung von sogenannten Wildcards. Diese Zeichen ersetzen entweder genau ein beliebiges Zeichen (jedoch nicht '\0'), dann nennt man sie Einzelquantoren, oder aber beliebig viele (auch null) beliebige Zei-chen - dann heißen sie eben Allquantoren. Sehr sinnvoll erweisen sich diese Wildcards in Verbindung mit der Suche nach bestimmten Dateien, wozu die GEMDOS-Funktion SFIRST (0x4E) die beiden Quantoren '?' und '*' bereitstellt. Ok, kennt jeder. Allerdings wird dabei ein wenig "gemogelt", weil der Dateiname in zwei Segmente aufgegliedert wird (Name und Extender). Steht nun in einem der Bereiche ein Sternchen, wird er ab dieser Stelle mit Fragezeichen aufgefüllt, egal ob hinter dem Stern noch etwas kommt oder nicht! Für das ST-Directory-System mag's ja genügen, in anderen Systemen wie UNIX sind da schon etwas diffizilere Abfragen nötig (Beispiel gefällig? Bitte: p*Doc*?_V0??). Abgesehen von Dateisystemen gibt es aber auch noch Textverarbeitung. Die Praxis hat gezeigt, daß es hin und wieder recht nützlich ist, eine Datei auf das Vorkommen einer Zeichenkette zu untersuchen. Hier läßt sich match() gewinnbringend zum Einsatz bringen. Der Patternstring (also der, in dem drinsteht, was gesucht werden soll) darf eine beliebige Kombination aus normalen Zeichen, All- und Einzelquantoren enthalten und muß nach dem Sourcestring (das ist der andere!) als zweiter Pointer übergeben werden. Der dritte Parameter ist die Adresse eines Stringpointers. Falls das Muster gefunden werden kann, enthält er die Adresse, ab der das letzte Muster gematcht werden konnte. Ein Beispiel: Sourcestring ="holladibolla", Patternstring=="*oll?*?*" gibt außer dem Ergebnis TRUE den Zeiger auf "dibolla" zurück. Das hat folgenden Vorteil: möchte man untersuchen, ob die Suchzeichenfolge nochmal vorkommt, muß man die Funktion match nur noch einmal mit dem um 1 erhöhten Stringzeiger aufrufen und schon wird richtigerweise FALSE zurückgegeben, weil durch die beiden Einzelquantoren noch mindestens zwei Zeichen hinter "oll" folgen müßten. Wäre dagegen beim ersten Aufruf "holladibolla" zurückgegeben worden, was sicherlich ein richtiges Ergebnis gewesen wäre, hätte ein zweiter Aufruf mit "olladibolla" wegen des führenden Allquantors (*) wieder ein "TRUE" ergeben, was sicherlich nichts weltbewegend Neues gebracht hätte. Übrigens kann man die beiden Quantoren beliebig umdefinieren (s. Demo), falls sie Bestandteil des Sourcestrings sein sollten!

## Kochrezept für "Match mit Quantor"

Zunächst einmal müssen die Stringlängen ermittelt werden, damit man auch weiß, wann man spätestens mit der Suche aufzuhören hat. Sollte mindestens einer der Strings leer sein, erübrigt sich natürlich die Suche. In einer Endlosschleife wird dann der String auseinandergenommen. Enthalten beide Strings ein normales Zeichen (also keinen Quantor), wird die Suche erfolglos beendet, wenn sie nicht übereinstimmen. Enthält der Patternstring einen Einzelquantor, muß im Sourcestring noch mindestens ein Zeichen vorhanden sein. Soweit, so hoopy. Lustig wird's erst beim Auftreten eines Allquantors. Angenommen, man sucht im String "Oh du zerfrettelter Grunzwanzling" das Muster "*nzl*". Läuft man sequentiell durch den String, findet man zuerst das "nz" von "Grunz". Das Problem ist, daß kein "l" folgt und also das Muster hier nicht greift. Wohl aber bei "wanzling". Schlimmer wird's, wenn Quantoren geschachtelt auftreten (**n*?l??*)! Man kann sich beliebig komplexe Suchzeichenfolgen ausdenken (die nur so von Quantoren strotzen), und ebenso fiese Sourcestrings, bei denen es einem vom bloßen Hinsehen schwindlig wird. Mit sequentiellen Algorithmen gerät man hier doch sehr schnell an die Grenzen des Abstraktionsvermögens! Viel einfacher ist in solchen Fällen die Verwendung von rekursiven Prozeduren. Betrachten Sie einmal, was passiert, wenn ein Allquantor im Muster auftaucht: Zunächst wird geprüft, ob es das letzte Zeichen im Muster ist. Falls ja, ist die Suche erfolgreich gewesen. Ansonsten muß das nächste Zeichen im String gesucht werden, das mit dem nächsten Zeichen des Musters (also dem hinter dem Allquantor) übereinstimmt. Ein Sonderfall tritt auf, wenn der String bereits vollständig durchsucht wurde, aber im Muster ein weiterer Allquantor auftaucht. Dieser matcht nämlich per Definition auch null Zeichen, so daß die Suche nicht einfach erfolglos abgebrochen werden darf! Andernfalls darf die Rekursion ihre Fähigkeiten demonstrieren: Falls es möglich ist, den String ab der aktuellen Position zu matchen, war die Suche erfolgreich; sonst wird es rekursiv einfach ab der nächsten Position versucht und zwar solange, bis der String komplett abgeprüft wurde (is>ls). In diesem Falle ist die Suche dann wirklich erfolglos. Das Schöne an der Rekursivität ist, daß sie immer zum richtigen Ergebnis gelangt; man braucht sich überhaupt nicht darum zu kümmern, wieso und weshalb - sie tut's einfach! Herrlich. Mittels dem printf()-Statement zu Beginn der Prozedur kann man die Arbeitsweise des Algorithmus bei Auftreten von Allquantoren nachvollziehen.

Match() ist sehr flexibel: Übergibt man ihr als Maske eine in Allquantoren eingebettete Zeichenkette (z.B. *blub*), verhält sie sich genau wie INSTR oder POS; läßt man den letzten Stern weg, so werden nur solche Strings gematcht, in denen die Maske am Ende auftaucht (wie bei "Lausblub"), und insgesamt kann man damit viel filigranere Dateisuchen veranstalten als mit SFIRST.

Fazit: Match! kann zwar nicht als Ersatz für die Query Language einer Datenbank dienen, erleichtert aber das Suchen von Dateien und Texten nicht unerheblich.

*MS*

*Listing folgt...*

```
 1: /**************************************************\
 2: *                 > > > M A T C H ! < < <          *
 3: *==================================================*
 4: * Univers.Suchfkt. z.Auffinden v.Substrings ü. Muster *
 5: * Sprache: MEGAMAX C  Autor: M. Schumacher   15.02.88 *
 6: \**************************************************/
 7: #include <stdio.h>
 8:
 9: match(what, how, where, all, one)
10:       char *what, *how, **where, all, one;
11: {
12:   char *s, *p;                       /* Stringpointer für
                                            what/how  */
13:   int  ls=0, lp=0, is=1, ip=1;  /* Stringlängen,
                                            Laufvariablen */
14:
15:   *where=what;  /* Falls s gemätscht werden kann, dann
                             ab hier */
16:   s=what; while (*s++) ls++; s=what; /* Stringlängen
                                               bestimmen */
17:   p=how;  while (*p++) lp++; p=how;
18:   if (!(ls*lp)) return(0);         /* Leerstring übergeben
                                            =>FAIL! */
19:   printf("Vergleiche: >%s< (%d) mit >%s< (%d):\n", s, ls,
         p, lp);
20:
21:   while (1)
22:   {  if (*p == one)
23:      {  if (is>ls) return(0);
24:         ++is; ++s; ++ip; ++p;
25:         if ( (is>ls) && (ip>lp) ) return(1);
26:         if (ip>lp) return(0);
27:         continue;
28:      }
29:
30:      if (*p == all)
31:      {  if (ip == lp) return(1);
32:         ++ip; ++p;
33:         if (is>ls) continue;
34:         do
35:         {  if (match(s, p, where, all, one)) return(1);
36:            ++is; ++s;
37:         }while (is<=ls);
38:         return(0);
39:      }
40:
41:      if ( (is>ls) || (!(*s++ == *p++))) return(0);
42:      ++ip; if ( (++is>ls) && (ip>lp) ) return(1);
43:   }
44: }
45:
46: main()
47: {
48:   char str[80], pat[80], *res, all, one;
49:
50:   printf("Teststring eingeben: "); gets(str);
51:   printf("Allquantor          : *%c", 8); all=getchar();
52:   if (all == '\n') all='*'; else getchar();
53:   printf("Einzelquantor       : ?%c", 8); one=getchar();
54:   if (one == '\n') one='?'; else getchar();
55:   while (1)
56:   {  printf("\n\nSuche nach: "); gets(pat); res=str;
57:      while (match(res, pat, &res, all, one))
58:      {  printf("TRUE: <%s>\n", res); res++; }
59:      puts("FAIL");
60:   }
61: }
```

***ENDE***

# ACCESSORIES NEU LADEN

**Manchmal möchte man die im System installierten Accessories entfernen und stattdessen andere Accessories installieren. Dabei hilft das hier vorgestellte Programm.**

Der Besitzer eines Farbmonitors hat es relativ einfach: Beim Umschalten zwischen mittlerer und hoher Auflösung werden alle installierten Accessories entfernt, nebenbei noch alle VDI-Workstations geschlossen und anschließend das AES neu initialisiert, was u.a. auch dazu führt, daß die Accessories von der gerade eingelegten Diskette neu geladen werden. Als Farbmonitorbesitzer schaltet man also zweimal die Auflösung um, und schon hat man das Ziel erreicht.

Der Besitzer eines Monochrommonitors hat es leider nicht so einfach. Doch wenn bei Umschaltung zwischen den Auflösungen das AES reinitialisiert wird, muß es einen legalen Weg geben, dies auch ohne Auflösungsumschaltung zu erreichen.

Erinnern wir uns daran, daß GEM vom IBM-PC auf den Atari ST portiert worden ist: Es gibt bei der IBM-Version die Möglichkeit, das GEM zu verlassen - auch dabei werden alle Accessories entfernt. Dann kann man GEM neu starten, und die Accessories werden neu geladen. Das Terminieren des GEM erreicht man durch spezielle Parameter der Funktion 'shel_write', die eigentlich dazu dient, andere Programme zu starten.

*shel_write(x, y, z, "PROGNAME.PRG", "command tail");*

Dabei sind "PROGNAME.PRG" der Name des zu startenden Programms und "command tail" die Parameter, die an dieses Programm zu übergeben sind. Wesentlich sind auch die ersten 3 Parameter, bei denen es sich um Integerzahlen handelt:

- z gibt an, ob es sich um eine GEM- oder TOS-Applikation handelt. 0 = TOS-Applikation, 1 = GEM-Applikation.
- y gibt an, ob es sich um eine grafische (1) oder um eine Textapplikation (0) handelt.
- x gibt an, ob das mit "PROGNAME.PRG" spezifizierte Programm ausgeführt (1) oder GEM terminiert werden soll (0).

Terminiert man das GEM auf dem PC, landet man wieder im normalen COMMAND.COM. Auf dem ST gibt es jedoch keinen Kommandointerpreter, vielmehr ist GEM die Standard-Shell.

Daher führt eine GEM-Terminierung zu einem GEM-Neustart, mit dem gewünschten Erfolg: Die Accessories werden neu geladen.

Doch wie so oft stimmt die Dokumentation mal wieder nicht mit der Software überein. Daher waren einige Experimente mit den Parametern x, y und z nötig, um das gewünschte Ergebnis zu erhalten. Doch es hat funktioniert, und das Resultat sehen Sie im Listing.

INITAES.C kann sowohl mit dem Entwicklungspaket als auch mit Megamax C kompiliert werden und läuft auf allen TOS-Versionen.

Die Accessories und das DESKTOP.INF werden immer von dem Laufwerk geladen, von dem INITAES.PRG gestartet wurde, es sei denn, es existiert ein Laufwerk C: im System (Ramdisk oder Harddisk); in diesem Falle werden Accessories und DESKTOP.INF von dort geladen.

Vorher installierte Ramdisks, Harddisktreiber etc. sind dann immer noch vorhanden. Vorsicht ist allerding geboten bei residenten Programmen, die vom Desktop aus gestartet wurden: Wegen der

reichlich fehlerhaften Speicherverwaltung des TOS ist nach der AES-Reinitialisierung u.U. erheblich weniger Speicher frei, weil die 'alten' Accessories dann noch als Leichen im Speicher liegen. Dies geschieht jedoch auch bei Farbmonitoren, wenn die Auflösung umgeschaltet wird. Es empfiehlt sich daher, residente Programme (Ramdisks, Harddisktreiber, Spooler etc.) immer aus dem AUTO-Ordner zu starten.

*Manfred Erprath*

```
 1:  /*
 2:   * INITAES.C reinitialize AES - load accessories and
                                     DESKTOP.INF
 3:   * (c) 1986 by Manfred Erprath
 4:   */
 5:
 6:  int        contrl[12];
 7:  int        intin[16],ptsin[16],intout[16],ptsout[16];
 8:
 9:  main()
10:  {
11:    appl_init();
12:    shel_write(0,0,-1,"","");    /* terminate AES */
13:    appl_exit();
14:  }
```

ENDE



# ACCESSORY AUCH ÜBER TASTATUR

**ST-Usern nicht unbekannt ist die Tatsache, daß die Accessories nur via Maus zu erreichen sind.**
**Mit einem einfachen Trick (das Wort Trick ist schon fast zu hoch gegriffen) ist es möglich, diese Accessories auch mit der Tastatur anzusprechen. Wenn man möchte, kann der Menüeintrag auch ganz entfallen.**

Keyboard-Events - auf einen Event reagiert ein Accessory - sprechen ein Accessory höchst selten an, also wird ein Umweg benutzt. Die Sonderfunktionstasten (Shift rechts und links, Alternate, Control, Maustaste rechts und links und CapsLock) haben einen Status, der sich abfragen läßt.
Unser Accessory wartet nun nicht nur auf einen Message-Event, der signalisieren würde, daß es angewählt wurde, sondern auf einen Timer-Event. Dieser Timer-Event wird alle 100 ms (Millisekunden) gegeben und das Accessory fragt alle 100 ms die Sondertasten ab. Sollte die von uns vorbestimmte Kombination gedrückt sein, so wird eine Aktion ausgelöst.
Diese 100 ms sind für den Programmbenutzer unmerklich, sowohl von der leichten Behinderung des Programmablaufes her als auch von der Wartezeit, bis die Aktion ausgelöst wird.
Das Accessory kann selbstverständlich nur dann aufgerufen werden, wenn ein Event möglich ist. Also ist ein Aufruf unter TOS-Anwendungen nicht möglich.
Was kann man alles aufrufen? Eigentlich alles! Das Beispiel-Accessory stellt nur eine Alert-Box dar. Die Routinen können aber sehr vielfältig sein. Damit keine Probleme mit anderen Programmen entstehen, sollte man aber möglichst die Wahl der Sondertasten, die das Accessory aktivieren, dem Programmbenutzer überlassen.
Die Abfrage der Sondertasten wird über die (Modula 2-)BIOS-Funktion GetKBShift bewerkstelligt. Im Bios selbst heißt die Funktion 'kbshift' und trägt die Nummer 11. Diese Funktion liefert bei Megamax Modula 2, mit dem das Accessory geschrieben wurde, einen Set mit den gedrückten Tasten zurück. Bei Pascal Plus beispielsweise kann nur ein Integer-Wert abgefragt werden. Man muß dort anhand der einzelnen Bits überprüfen, welche Taste gedrückt wurde.
Das Modul BIOS aus dem Megamax Modula 2-Modul stellt einen Set mit folgenden Tasten zur Verfügung:

| | |
|---|---|
| Shift rechts | : RightShift |
| Shift links | : LeftShift |
| Control | : ControlKey |
| Alternate | : AlternateKey |
| Caps Lock | : CapsLock |
| Maustaste rechts | : RightMouseButton |
| Maustaste links | : LeftMouseButton |

Bei dem Beispiel-Accessory wird geprüft, ob beide Shift-Tasten gedrückt sind. Für andere Tastenkombinationen sind entsprechend andere Werte abzufragen. Man sollte aber möglichst Doppelbelegungen verwenden, da eine Aktion etwa auf Shift hin nicht unbedingt wünschenswert ist.
Möglichkeiten für eigene Programme gibt es reichlich. Es ist ja auch möglich, auf eine bestimmte Tastenkombination hin eine Dialog-Box zu öffnen und dort verschiedene Tätigkeiten zur Auswahl zu stellen.

*Dietmar Rabich*

*Literatur: Atari ST Profibuch, Jankowski/Reschke/Rabich, Sybex-Verlag*

*Listing folgt...*

```
 1: (**************************************************)
 2: (* Accessory, aufrufbar über Desk-Eintrag und
      Tastatur 2.0  *)
 3: (* ---------------------------------------------- *)
 4: (* Entwickelt mit Megamax Modula 2 von Application  *)
 5: (* Systems, Heidelberg.                              *)
 6: (* Autor: Dietmar Rabich, Dülmen                     *)
 7: (* Datum: 19. März 1988                              *)
 8: (**************************************************)
 9:
10: MODULE HideACC;
11:
12: (*$S1*) (* <- reservierter Stack. Für andere  *)
13:         (*    Anwendungen ggf. zu ändern.     *)
14:
15: (* Importe *)
16: FROM GrafBase   IMPORT Point,Rect;
17: FROM AESEvents  IMPORT MultiEvent,Event,EventSet,
                           MessageBuffer,
18:                        accOpen,RectEnterMode;
19: FROM AESForms   IMPORT FormAlert;
20: FROM AESMenus   IMPORT RegisterAcc;
21: FROM GEMEnv     IMPORT RC,InitGem,DeviceHandle;
22: FROM GEMGlobals IMPORT MButtonSet,SpecialKeySet,GemChar,
23:                        PtrMaxStr;
24: FROM BIOS       IMPORT KBShifts,KBShiftBits,GetKBShift;
25: FROM Storage    IMPORT ALLOCATE;
26:
27: (* Zeit für Timer *)
28: CONST repeatTime = 100L;
29:
30: (* Variablen *)
31: VAR ProgramName  : PtrMaxStr;
32:     OK           : BOOLEAN;
33:     Events       : EventSet;
34:     Message      : MessageBuffer;
35:     dummyPoint   : Point;
36:     dummyBut     : MButtonSet;
37:     dummyChar    : GemChar;
38:     dummyClick,
39:     dummyCard,
40:     ApplID       : CARDINAL;
41:     specialKeys  : SpecialKeySet;
42:     dev          : DeviceHandle;
43:     KeyKomb      : KBShifts;
44:
45: (* Unsere Beispielanwendung *)
46: PROCEDURE MachWas;
47:
48:  BEGIN
49:   FormAlert(1,'[0][Hello World!][***]',dummyCard)
50:  END MachWas;
51:
52: BEGIN
53:  ALLOCATE(ProgramName,256);
54:  ProgramName^:='  Accessory';    (* Accessoryname *)
55:  InitGem(RC,dev,OK);
56:
57:  (* Tastenkombination, die den Aufruf ermöglicht *)
58:  KeyKomb:=KBShifts{RightShift,LeftShift};
59:
60:  IF OK THEN
61:    RegisterAcc(ProgramName,ApplID,OK);
62:    IF OK THEN
63:     REPEAT
64:      MultiEvent (EventSet{message,timer},
65:                  0,MButtonSet{},MButtonSet{},
66:                  lookForEntry,Rect(0,0,0,0),
67:                  lookForEntry,Rect(0,0,0,0),
68:                  Message,
69:                  repeatTime,
70:                  dummyPoint,dummyBut,specialKeys,
                     dummyChar,
71:                  dummyClick,Events);
72:
73:        (* Aktion über Menüeintrag    *)
74:      IF ((message IN Events) AND (Message.msgType =
         accOpen)) OR
75:        (* Aktion über Tastaturaufruf *)
76:        ((timer IN Events) AND (KeyKomb*GetKBShift()=
          KeyKomb))
77:
78:       THEN
79:       EXCL(Events,message);
80:       MachWas
81:      END;
82:
83:     UNTIL FALSE
84:    ELSE
85:     FormAlert(1,'[0][Kein Accessorystart möglich][Ende]',
86:               dummyCard)
87:    END
88:  END
89: END HideACC.
90:
91: (* - - - - - - - Ende des Listings  - - - - - - - *)
```

*ENDE*

# Betr.: Programmier Praxis

Die "Progammier Praxis" ist ein fester Bestandteil der ST Computer geworden. Gedacht ist sie für all diejenigen, die ihren Rechner selbst programmieren, und dazu Tips und Anregungen gebrauchen können. Diese Rubrik kann aber nur dann bestehen, solange viele Leser (Sie eingeschlossen) sich daran beteilligen. Wir fordern Sie deshalb auf, Ihre Ideen in einen Umschlag zu stecken und auf dem schnellsten Weg zu uns zu schicken. Voraussetzungen gibt es praktisch keine: es muß lediglich interessant sein und so dokumentiert, daß auch andere Anwender daraus einen Nutzen ziehen können. Auch bei den Sprachen gibt es keinerlei Einschränkungen: PASCAL, C, BASIC; Assembler, Modula 2, APL, Fortan, LISP, Prolog, PEARL,... und sogar LOGO ist erwünscht.
Veröffentliche Programme werden natürlich angemessen honoriert. Bitte beachten Sie bei der Einsendung: Schicken Sie den Quelltext und das evtl. compilierte Programm, sowie die Dokumentation auf Papier und Diskette. Die Diskette wird zurückerstattet.
Einsendungen direkt an die Redaktion:

**"MERLIN" Computer GmbH**
**ST Computer Redaktion**
**´Programmier Praxis´**
**Industriestraße 26**
**D-6236 Eschborn Tel.: 0 61 96 / 48 18 11**

# SAG MIR DEINE NAMEN, BS-HANDEL!

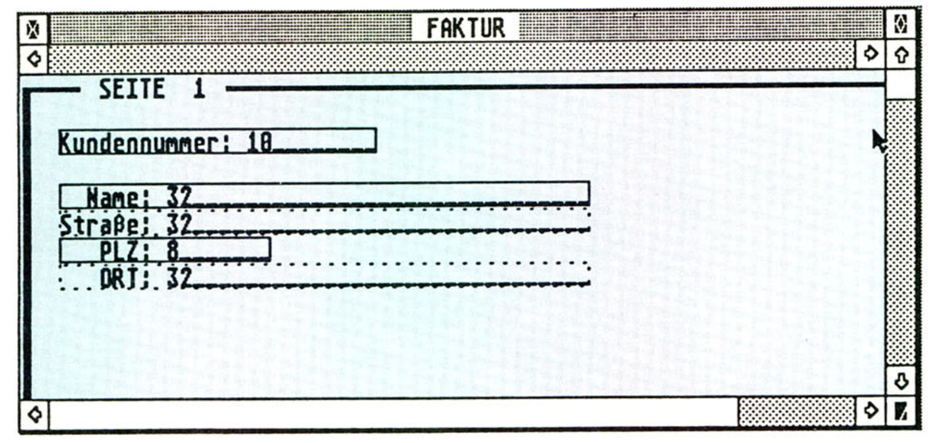



*Bild 1: Adimens-Maske für BS-Handel-Adreßdaten.*

**Bei der Fakturierung mit BS-Handel gibt man im Laufe der Zeit sehr viele Adressen ein. Wenn diese Daten nach BS-Handel weiterverarbeitet werden sollen, müssen sie aus der Adreßdatei von BS-Handel herausgeholt werden. Am sinnvollsten ist momentan auf dem Atari ST wohl eine Weiterbearbeitung mit der Datenbank ADIMENS. Im Folgenden wird der Ablauf für eine Konvertierung beschrieben, und ein kleines Programm in C erledigt die notwendige Arbeit, die Adreßdaten aus BS-Handel in eine Export (.EXP) Datei für Adimens zu wandeln. Das Programm arbeitet in seiner derzeitigen Version mit dem Adimens 2.0 zusammen.**

## Raus aus dem BS-Handel

Zuerst muß das C-Programm eingetippt und mit dem Megamax C-Compiler compiliert werden (andere C-Compiler tun es auch).

Das fertig compilierte Programm sollte dann auf die gleiche Partition gespielt werden, auf der sich die Datei ADRESSEN.DAT von BS-Handel befindet. Vor dem Starten des Programms sollte gewährleistet sein, daß sich auf der Partition noch genügend freier Speicherplatz für die .EXP Datei befindet. Die Länge der vom Programm erzeugten Datei für Adimens beträgt ca. ein Zehntel der BS-Handel-Adreßdatei. Nach dem Starten des Programms werden alle Datensätze konvertiert, und die aktuelle Nummer wird links oben im Bildschirm heraufgezählt. Wenn alle Datensätze konvertiert sind, erscheint die Anzahl der insgesamt gefundenen Adressen.

## Rein ins Adimens

Wenn alles geklappt hat, sollte mit Adimens eine Datei FAKTUR mit Maske erstellt werden, die die Merkmale KUNDENNUMMER, NAME, STRASSE, PLZ und ORT enthält.

Die einzelnen Datenfelder sollten als Texte definiert sein, und zur Weiterverarbeitung empfiehlt sich die Benutzung von Schlüsseln für Kundennummer, Name und Postleitzahl. Eine Maske, die sehr schnell mit Adimens zu erstellen ist, könnte wie in Bild 1 aussehen.

Bei der Erstellung der Maske im Adimens-INIT sollte unbedingt beachtet werden, daß bei der Einstellung der Größe der Datei mindestens die Anzahl der gefundenen Adressen eingegeben wird. Sicherheitshalber sollte großzügig nach oben aufgerundet werden. Nachdem die Maske fertiggestellt ist, kann im Hauptteil von Adimens die zuvor erstellte Datei importiert werden.

*OJO*

```
 1:  /*
 2:  Programm zum Wandeln der Adressdatensaetze von BS-Handel
 3:  2.02 in eine Adimens Datei. Die Maske der Adimens Datei
 4:  sollte die Eintraege
 5:
 6:  KUNDENNUMMER, NAME, STRASSE, PLZ, ORT
 7:
 8:  in der Maske besitzen. Dabei ist eine Wahl der Felder
 9:  als Text mit jeweils ca. 30 Zeichen sinnvoll. Fuer eine
10:  Weiterverarbeitung sollten die Eintraege KUNDENNUMMER,
11:  NAME und PLZ als Schluesselwoerter verwendet werden.
12:  Die Datei, die mit diesem Programm erstellt wird enthaelt
13:  eine komplette Exportliste fuer Adimens.
14:
15:  Programmautor:   Oliver Joppich
16:  Programmversion: Megamax C, 08.05.88
17:
18:  */
19:
20:
21:  /* Standard Include Dateien */
22:  #include <osbind.h>
23:
24:
25:  /* Konstanten Offsets fuer BS-Handel Adresssaetze */
26:  #define NUMMER  0
27:  #define NAME    12
28:  #define STRASSE 0x4e
29:  #define PLZ             0x6d
30:  #define ORT             0x77
31:  #define TELEFON 0x8c
32:
```

```
33:
34: /* Puffer zum Einlesen eines Datensatzes */
35: char buffer[0x500];
36:
37:
38:
39: main()
40: {
41:         int i;
42:         int fh,fh2;
43:         long eof;
44:
45:         fh = Fopen("ADRESSEN.DAT",0);
                        /* Name der BS-Handel Datei */
46:         if (fh >0)
47:         {
48:                 Fseek(0x21fL,fh,0);
                        /* an den Datenanfang */
49:                 fh2     = Fcreate("FAKTUR.EXP",0);
                        /* Adimens Export Datei */
50:                 if (fh2 >0)
51:                 {
52:                         Fw_string(fh2,".FILE FAKTUR");
                            /* Kennung fuer Adimens */
53:                         Fw_crlf(fh2);
                            /* Start mit Returns */
54:                         Fw_crlf(fh2);
55:                         Fw_crlf(fh2);
56:
57:                         while (Fread(fh, 0x342L,
                            buffer) >0L )
                            /* solange Daten schreiben  */
58:                                 writedata(fh2);
                            /* bis Ende erreicht ist. */
59:
60:                         Fclose(fh2);
61:                 }
62:                 Fclose(fh);
63:         }
64:         printf("Datensaetze gefunden ...\n");
65:         Cconin();
66: }
67:
68: writedata(fh2)
69: int fh2;
70: {
71:         static int count = 0;
72:
73:         printf("%cH %6d\n",0x1b,count++);
            /* NUMMER des Datensatzes zur Kontrolle */
74:
75:         Fw_string(fh2, &buffer[NUMMER]);
            /* Schreiben der Kundennummer */
76:         Fw_string(fh2, &buffer[NAME]);
            /* Schreiben des Namen */
77:         Fw_string(fh2, &buffer[STRASSE]); /* Strasse */
78:         Fw_string(fh2, &buffer[PLZ]); /* Die Postleitzahl */
79:         Fw_string(fh2, &buffer[ORT]);
            /* Der ... (na was wohl ?) */
80:         Fw_crlf(fh2);
81: }
82:
83: Fw_string(fh, s)
84: int fh;
85: char *s;
86: {
87:         Fwrite(fh, (long)strlen(s), s);
88:         Fw_crlf(fh);
89: }
90:
91: Fw_crlf(fh)
92: int fh;
93: {
94:         char crlf[2];
95:
96:         crlf[0] = 0x0d;
97:         crlf[1] = 0x0a;
98:
99:         Fwrite(fh, 2L, crlf);
100: }
```

*ENDE*

# DAS GRÖßERE DISKETTENFORMAT

**Wem ist es wohl noch nicht passiert, Dateien auf eine Diskette kopieren zu wollen und dann plötzlich festzustellen, daß der Platz nicht reicht? Bisher gab's meist nur zwei Möglichkeiten: Neuformatieren mit größerem Format oder eine zweite Diskette zur Hand zu nehmen.**

Dabei handelt es sich meist nur um ein paar KiloBytes. Schöner wäre es doch, wenn die eingelegte Diskette plötzlich mehr Platz hätte! Aber, Neuformatieren hat das Löschen der Dateien zur Folge. Nicht so mit dem hier vorgestellten Diskettenformatierer. Das Programm formatiert einfach zwei unbenutzte Tracks (Nummer 80 und 81) und läßt alle anderen in Ruhe. Damit man über den Platz auch verfügen kann, wird der Bootsektor angepaßt. Das bringt pro Diskettenseite immerhin 10 kB ohne größere Umstände! Das Programm wird gestartet und untersucht ohne weitere Nachfrage die Diskette in Laufwerk A. Ist sie normal formatiert (360 oder 720 kB), werden die zusätzlichen Tracks formatiert, sofern kein Fehler auftritt. Sollte das Format nicht stimmen, wird dies mit einer Fehlermeldung zum Ausdruck gebracht. Das Programm stellt selbst fest, welches Diskettenformat vorliegt. Der Benutzer braucht also nur die passende Diskette in Laufwerk A zu legen und dann das Formatierprogramm zu starten.

In der Kürze liegt die Würze. Das war's schon mit all den Erklärungen! Zum Programm selbst steht noch etwas im (Pascal Plus-)Listing! Ich wünsche Ihnen viel Freude und immer ausreichend Platz auf allen Disketten!

*Dietmar Rabich*

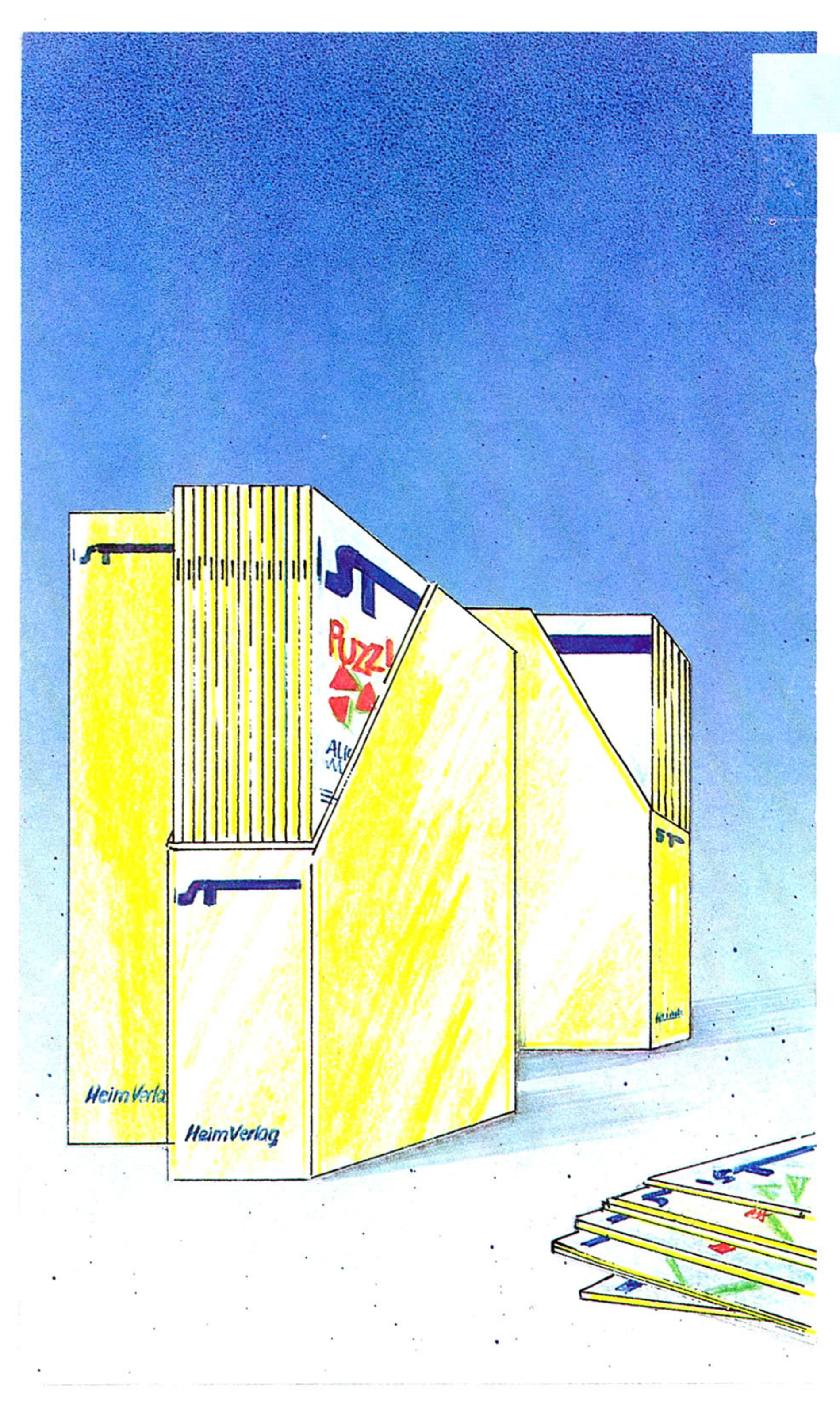
```
1:  (* Programm FORMAT PLUS *)
2:  program formatplus;
3:
4:  (* Entwickelt mit ST Pascal Plus 2.0.          *)
5:  (* 16. Dezember 1987 / Dietmar Rabich          *)
6:  (* Literatur: Atari ST Profibuch (Sybex 1987)  *)
7:
8:  type buffer_klein  = packed array [0..511] of byte;
    (* 0.5 KByte *)
9:
10: var bootsekt,filler_kl  : buffer_klein;
11:     seiten              : short_integer;
12:
13: (* Liest Sektor *)
```

```
14:  procedure floprd(var buf,filler   : buffer_klein;
15:                   devno,sectno,trackno,sideno,count :
                                          short_integer);
16:   XBIOS(8);
17:
18:  (* Schreibt Sektor *)
19:  procedure flopwr(var buf,filler   : buffer_klein;
20:                   devno,sectno,trackno,sideno,count :
                                          short_integer);
21:   XBIOS(9);
22:
23:   (* Formatiert Track 80 und 81 *)
24:   function format (side : short_integer) : boolean;
25:
26:    type buffer_gross  = packed array [0..8191] of byte;
    (* 8 KByte *)
27:
28:    var i,error_f   : short_integer;
29:        buf,filler : buffer_gross;
30:
31:    (* Formatiert einen Track *)
32:    function flopfmt(var buf,filler : buffer_gross;
33:                     devno,spt,trackno,sideno,interlv :
                                          short_integer;
34:                     magic           : long_integer;
35:                     virgin          : short_integer)
36:                     :short_integer;
37:     XBIOS(10);
38:
39:    begin
40:     i:=80;
41:     repeat
42:      error_f:=flopfmt(buf,filler,0,9,i,side,1,$87654321,
                                                  $0000);
43:      i:=i+1;
44:     until (error_f<>0) or (i>=82);
45:     format:=(error_f<>0);
46:    end;
47:
48:   (* Fehlerausgabe *)
49:   procedure fehler(erg:boolean);
50:
51:    begin
52:     if erg then
53:      begin
54:       writeln('*** Fehler bei Formatiervorgang / falsches
                                              Format ***');
55:        repeat
56:        until keypress;
57:        halt
58:       end;
59:     end;
60:
61:   (* Hauptprogramm *)
62:   begin
63:    writeln('*** Format Plus / (c) Dietmar Rabich ***');
64:    writeln('  (Entwickelt mit ST Pascal Plus 2.00.)');
65:    floprd(bootsekt,filler_kl,0,1,0,0,1); (* Bootsektor
                                             lesen *)
66:    if (bootsekt[19]=$A0) and (bootsekt[20]=$05) then
                                 (* zweiseitig (720 kB) ? *)
67:     seiten:=2;
68:    if (bootsekt[19]=$D0) and (bootsekt[20]=$02) then
                                 (* einseitig (360 kB) ?  *)
69:     seiten:=1;
70:    if seiten in [1,2] then
71:     begin
72:      fehler(format(0));          (* formatiere Seite 1 *)
73:      if seiten=2 then
74:       fehler(format(1));         (* formatiere Seite 2 *)
75:      if seiten=2 then            (* neue Bootsektoren  *)
76:       begin
77:        bootsekt[19]:=$C4;        (* 1476 Sektoren insgesamt *)
78:        bootsekt[20]:=$05;
79:       end
80:      else
81:       begin
82:        bootsekt[19]:=$E2;        (* 738 Sektoren insgesamt  *)
83:        bootsekt[20]:=$02;
84:       end;
85:      flopwr(bootsekt,filler_kl,0,1,0,0,1); (* Bootsektor
                                                schreiben *)
86:     end
87:    else
88:     fehler(true);
89:   end.
90:   (* Programmende *)
```

*ENDE*



# BRINGEN SIE ORDNUNG IN IHRE ATARI ST ZEITSCHRIFTEN-SAMMLUNG MIT DEN PRAKTISCHEN SAMMEL-BOXEN!




**VERGESSEN SIE**

- Gestapelte, verrutschte und gekippte Zeitschriften
- Mühseliges Suchen nach einer bestimmten Ausgabe

**NUTZEN SIE DIE VORTEILE DER NEUEN SAMMELBOXEN**

- Übersicht
- Schneller und leichter Zugriff auf gewünschte Ausgaben
- Platzsparende und geschützte Aufbewahrung Ihrer Sammlung
- Ihre ST-COMPUTER-Zeitschriften werden zu einem geordneten und wertvollen Nachschlagewerk

**BESTELLEN SIE:**

- mit untenstehendem Bestellabschnitt
- Eine Box kostet DM 14,– zuzügl. einer Versandkostenpauschale von DM 3,–, unabhängig von der bestellten Stückzahl
- Die Auslieferung Ihrer Bestellung wird nur gegen Vorauszahlung durchgeführt
- Zahlen Sie bitte mit Scheck oder durch Voraus-Überweisung unter genauer Angabe Ihrer Wünsche.
- Die Auslieferung erfolgt sofort nach Zahlungseingang

**ÜBRIGENS:**

- Mit den neuen ST-COMPUTER-Sammelboxen oder mit einem Geschenk-Abonnement von ST-COMPUTER machen Sie Freunden und Bekannten große Freude.

*unverbindlich empfohlener Verkaufspreis*

Benutzen Sie untenstehenden Bestellabschnitt und die Bestellkarten in der Zeitschrift.

## BESTELLCOUPON

Bitte senden Sie mir ________ Stück Sammelboxen ST COMPUTER zum Preis von DM 14,– je Stück zuzüglich DM 3,– Versandkosten unabhängig von der Bestellmenge.

Zahlung: ☐ per Nachnahme ☐ Scheck liegt bei ☐ per Vorausüberweisung

Name, Vorname

PLZ, Ort — Straße, Nr.

Datum — Unterschrift

**Heim Verlag**

Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

# *Einkaufsführer*

## *Hier finden Sie Ihren Atari Fachhändler*

### 1000 Berlin



**alpha computers g.m.b.h.**
u. a. alphatronic, atari, commodore, dai, epson, sord mit pips, nec hard-/software nach maß — servicetechnik
Kurfürstendamm 121a, 1000 Berlin 31 (Halensee)
Telefon 030/8911082

**Computare**
Keithstr. 18-20 • 1000 Berlin 30
030/21 390 21
186 346 com d

**DATAPLAY**
Bundesallee 25 · 1000 Berlin 31
Telefon: 030/861 91 61







Büroelektronik
Keithstraße 26 · 1000 Berlin 30
☎ 26 111 26

### 1000 Berlin




**ATARI**
... wir machen Spitzentechnologie preiswert.
Vertragshändler
**UNION ZEISS**
Kurfürstendamm 57 • 1000 Berlin 15
Telefon 32 30 61

**Computershop Behrendt**
Reinecke, Tscheuschner GbR
Fürbringerstraße 26 · 1000 Berlin 61
Tel. (0 30) 6 91 76 66 · BTX (030) 6 91 76 66



*Erfolgreich werben*
Sprechen Sie mit uns.
Heim-Verlag ☎ (0 61 51) 5 60 57 BUF

COMPUTER-STUDIO
**Schlichting**
... die etwas andere Computerei
ATARI-Fachmarkt
MS-DOS Fachmarkt · NEC-Fachhandel
Katzbachstraße 8 · 1000 Berlin 61
☎ 030/7 86 43 40

### 2000 Hamburg

**Bit Computer Shop**
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/49 44 00
**Createam**
Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon Sa. Nr. 040/641 50 91

Hardware
Software
Beratung
Service




ATARI Systemfachhändler
Münsterstraße 9 · 2000 Hamburg 54
Telefon 040/56 60 1-1

**Gerhard u. Bernd Waller GbR**
Computer und Zubehör-Shop
Kieler Straße 623
2000 Hamburg 54
Telefon (0 40) 5 70 60 07 + 5 70 52 75

**RADIX Bürotechnik**
Heinrich-Barth-Straße 13
2000 Hamburg 13
Telefon (0 40) 44 16 95
*NEU: Software Shop*




### 2000 Norderstedt




### 2120 Lüneburg

**Sienknecht**
**Bürokommunikation**
Beratung - Verkauf - Werkstatt
Heiligengeiststr. 20, 2120 Lüneburg
Tel. 0 41 31 / 4 61 22, Btx 40 24 22
Mo.-Fr. 9⁰⁰-18⁰⁰ und Sa. 9⁰⁰-13⁰⁰

## 2210 Itzehoe

**Der Computerladen**

Inhaber Ulrich Bübel · Martin Kopplow

Coriansberg 2 · 2210 Itzehoe
Telefon (0 48 21) 33 90 / 91

## 2300 Kiel




**Die Welt der Computer**
Dreiecksplatz Nr. 7
**2300 Kiel 1 · ☎ 04 31 / 56 70 42**

## 2350 Neumünster




## 2390 Flensburg




Norderstraße 94-96 · D-2390 Flensburg
☎ (04 61) 2 81 81 + 2 81 93

## 2800 Bremen




Faulenweg 48—52
2800 Bremen 1
Telefon (04 21) 17 05 77

## 2850 Bremerhaven

HEIM- UND PERSONALCOMPUTER



Kurt Neumann
„Bürger" 160
2850 Bremerhaven
Tel. 04 71 / 4 20 06

HARDWARE · SOFTWARE · PAPIERWARE

## 2940 Wilhelmshaven

**Radio Tiemann**
ATARI-Systemfachhändler

Markstr. 52
2940 Wilhelmshaven
Telefon 0 44 21 - 2 61 45

## 2950 Leer




- HARDWARE-SOFTWARE
- SYSTEM-ENTWICKLUNG
- ORGANISATION
- EDV-SCHULUNG
- EDV-BERATUNG
- SERVICE-WARTUNG

Augustenstraße 3 · 2950 Leer
Telefon 04 91 - 45 89

## 3000 Hannover

**COM DATA**

Am Schiffgraben 19 · 3000 Hannover 1
Telefon 05 11 - 32 67 36




**DATALOGIC COMPUTERSYSTEME**
ATARI ST- BERATUNG
COMPUTER SERVICE
HARDWARE VERKAUF
SOFTWARE
CALENBERGER STR. 26
3000 HANNOVER 1
TEL.: 0511 - 32 64 89

**trendDATA Computer**
IBM · EPSON · TRIUMPH ADLER
HEWLETT PACKARD · ATARI etc.

**trendDATA Computer GmbH**
**Am Marstall 18-22 · 3000 Hannover 1**
**Telefon (05 11) 1 66 05-0**

## 3040 Soltau

**F & T Computervertrieb**

Am Hornberg 1
(Industriegeb. Almhöhe)
3040 Soltau
Tel. 0 51 91 / 1 65 22

## 3150 Peine

**Wieckenberg & Schrage GmbH**
**Computertechnik**
**Hard- u. Software**

Woltorfer Str. 8, 3150 Peine
**Tel. 0 51 71 / 60 52/3 o. 0 51 73 / 79 09**

## 3170 Gifhorn

COMPUTER-HAUS
GIFHORN

Braunschweigerstr. 50
3170 Gifhorn
Telefon 0 53 71 - 5 44 98

## 3300 Braunschweig

**COMPUTER STUDIO**
**BRAUNSCHWEIG**

Rebenring 49-50
3300 Braunschweig
Tel. (05 31) 33 32 77 / 78

## 3400 Göttingen

Büroeinrichtungs-Zentrum




3400 Göttingen-Weende
Wagenstieg 14 – Tel. 05 51 / 38 57-0

## 3470 Höxter




## 3500 Kassel

**Hermann Fischer GmbH**
**autorisierter ATARI-Fachhändler**

Rudolf-Schwander-Str. 5-13
3500 Kassel
Telefon (05 61) 70 00 00

## 4000 Düsseldorf

**BERNSHAUS GmbH**
**Bürotechnik – Bürobedarf**

Cäcilienstraße 2
4000 Düsseldorf 13 (Benrath)
Telefon 02 11 - 71 91 81

## 4000 Düsseldorf

**HOCO**
**EDV ANLAGEN GMBH**
Flügelstr. 47
4000 Düsseldorf
Tel. 02 11 - 77 62 70

## 4050 Mönchengladbach




Hindenburgstr. 249
4050 Mönchengladbach
Tel. 0 21 61 - 187 64

## 4130 Moers




- Service-Center
- ATARI Fachhändler
- Hardware
- Software
- Erweiterungen

**COP Computer Service GmbH**
Essenberger Straße 2 H · 4130 Moers
Telefon (0 28 41) 235 85

## 4150 Krefeld




- Festplatten
- Scanner
- Drucker
- BTX-Module
- Literatut
- Zubehör

**COP Computer Service GmbH**
Lewerentz-Straße 111 · 4150 Krefeld
Telefon (0 21 51) 77 30 42

## 4250 Bottrop

**Megateam-Computer-Systeme**
Kirchhellenerstraße 262
4250 Bottrop

## 4300 Essen

ATARI Systemfachhändler




KARSTADT Aktiengesellschaft
Limbecker Platz 4300 Essen 1
Tel.: (02 01) 17 63 99

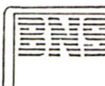
## 4320 Hattingen

Ihre Tür zur Zukunft:




Hattingen, Große Weilstr. 18-20, Telefon (0 23 24) 20 94 77

## 4330 Mülheim




Computer und Bürotechnik
Vertriebsgesellschaft mbH
Dickswall 79 4330 Mülheim Telefon 02 08/3 40 34

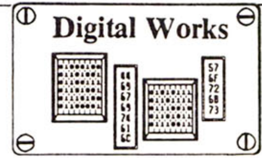
Tandon Computer
Computer Hard- und Software auch im Leasing
Computerkurse für Anfänger und Fortgeschrittene
VICTOR
SEL-Fernkopierer
NEC ATARI OKI

## 4400 Münster

**BASIS**
COMPUTER SYSTEME GMBH
Daimlerweg 39 - 4400 Münster
Telefon 02 51 / 71 99 75 - 9



## 4410 Warendorf




Computer-Fachhandel – Hardware & Software
Jörg Kellert – Helmut Müller GbR
Brünebrede 17 · 4410 Warendorf
Tel. 0 25 81 / 6 11 26

## 4422 Ahaus

ATARI · Epson · Fujitsu Molecular · NCR · Tandon · Schneider · Star
**OCB**

OCB-Computershop
Wallstraße 3
4422 Ahaus
Tel. 0 25 61/50 21

OCB-Hard- und Software
Wessumerstraße 49
4422 Ahaus
Tel. 0 25 61/50 21

## 4430 Steinfurt

ATARI SCHNEIDER STAR NEC SEIKOSHA PANASONIC EPSON
Computer
Büromaschinen
Service
Telefon 02551/2555
Tecklenburger Str. 27 · 4430 Steinfurt

## 4500 Osnabrück

**Heinicke-Electronic**
Kommenderiestr. 120 · 4500 Osnabrück
Telefon 05 41 - 8 27 99
Wir liefern Micro-Computer seit 1978

## 4600 Dortmund

**Bürostudio BOLZ**
Brauhausstraße 4 · 4600 Dortmund
Telefon 02 31 - 52 77 13 - 16

## 4600 Dortmund

ATARI Systemfachhändler




KARSTADT Aktiengesellschaft
Kampstraße 1 · 4600 Dortmund
Telefon (02 31) 5 43 91



Atari, Genie, Schneider, Tandy, Brother, Star, Memorex, BASF, Verbatim
cc Computer Studio GmbH
Software-Hardware-Beratung
Service-Eilversand
Ihre Ansprechpartner: v. Schablinski, Jan P. Schneider
Elisabethstraße 5
4600 Dortmund 1
T. 0231/528184 · Tx 822631 cccsd

**Elektronik**
**Computer**
**Fachliteratur**
**ATARI-System-Fachhändler**
4600 Dortmund 1, Güntherstraße 75, Tel. (02 31) 57 22 84




## 4620 Castrop-Rauxel

***R. Schuster Electronic***
OBERE MÜNSTERSTR. 33-35 · (0 23 05) 3 77 0 · 4620 CASTROP-RAUXEL
ATARI
System-Fachhändler

## 4650 Gelsenkirchen-Horst




Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel
Poststr. 15 · 4650 Gelsenkirchen-Horst
Tel. 0209/52572

## 4700 Hamm

computer center
Büromaschinen, Büroorganisation, Büromöbel, Werkstatt/Service
Moderne Büroausstattung
RÜTER
BÜRO RÜTER
Heinrich Rüter GmbH & Co. KG
4700 Hamm 1
Gustav-Heinemann-Straße 19/21
Telefon 02381/1404-0

## 4800 Bielefeld

hardware software organisation service
**CSF**
CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106 – 108
4800 Bielefeld 1
Tel. (05 21) 6 16 63

## 4800 Bielefeld

Carl-Severing-Str. 190
4800 Bielefeld 14

Telefon: 05 21/45 99-150
Telex : 9 37 340 krab d
Telefax: 05 21/45 99-123

MICROTEC
Software
Hardware
Beratung
Service

## 5000 Köln




AM RUDOLFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
TEL. (02 21) 21 91 71

## 5010 Bergheim




EDV-Beratung · Organisation
Programmierung · Home/Personal-Computer
Software · Zubehör · Fachliteratur

Zeppelinstr. 7 · 5010 Bergheim
Telefon 0 22 71 - 6 20 96

## 5090 Leverkusen

Rolf Rocke
Computer-Fachgeschäft
Auestraße 1
5090 Leverkusen 3
Telefon 0 21 71 / 26 24

## 5220 Waldbröl




Waldstraße 1 · 5220 Waldbröl
☎ (0 22 91) 44 08/33 86

## 5300 Bonn



Gesellschaft für Computer- und Communicationstechnologie mbH
Hardware · Software · EDV-Zubehör
Telefon 02 28/22 24 08
COCO GmbH · Schumannstraße 2 · 5300 Bonn 1

## 5457 Straßenhaus

**DR. AUMANN GMBH**
**Computer-Systeme**
Schulstr. 12
5457 Straßenhaus
Telefon 0 26 34 - 40 81/2

## 5500 Trier




Güterstr. 82 - 5500 Trier
☎ 06 51 - 2 50 44
Fordern Sie unsere Zubehör-Liste an.

## 5540 Prüm

**ATC** COMPUTER
J. M. ZABELL
Ritzstraße 13 · Pf. 10 51
**5540 PRÜM**
– Tel.: 0 65 51 - 30 39 –

## 5600 Wuppertal

**Jung am Wall**

Wall 31—33
**5600 Wuppertal 1**
Telefon 02 02/45 03 30

**COMPUTER FINKE**




ATARI – SYSTEMFACHHÄNDLER
KIPDORF 22 * 5600 WUPPERTAL 1 * TEL. 0202 - 45 32 33
HARDWARE * SOFTWARE * ZUBEHÖR * SERVICE * SCHULUNGEN
ATARI
... wir machen Spitzentechnologie preiswert.

**MEGABYTE**
Computer Vertriebs GmbH

Friedrich-Engels-Allee 162
5600 Wuppertal 2 (Barmen)
Telefon (02 02) 8 19 17

## 5630 Remscheid

**COM SOFT**
**Nordstraße 57 · 5630 Remscheid**
**Telefon (0 21 91) 2 10 33**

## 5650 Solingen

**MegaTeam**
Hardware — Software
Zubehör — Service

Rathausstraße 1-3 · 5650 Solingen 1
Telefon (02 12) 4 58 88

## 5800 Hagen




**Vertragshändler Axel Böckem**
**Computer + Textsysteme**
Eilper Str. 60 (Eilpezentrum) · 5800 Hagen
Telefon (0 23 31) 7 34 90

## 5900 Siegen




**Siegen · Weidenauer Str. 72 · ☎ 02 71/7 34 95**

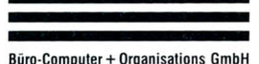
## 6000 Frankfurt

Müller & Nemecek
Kaiserstraße 44
6000 Frankfurt/M.
Tel. 0 69 - 23 25 44

**WAIZENEGGER**
Büroeinrichtungen
Kaiserstraße 41
6000 Frankfurt/Main
Tel. (0 69) 2 73 06 - 0



***Eickmann Computer***
***Die Profis***
***Beratung, Service, Zubehör***
In der Römerstadt 249
6000 Frankfurt/Main 90-Praunheim
Telefon (0 69) 76 34 09







Oederweg 7-9
6000 Frankfurt/Main 1
☎ (0 69) 55 04 56 - 57

Commodore OKI ATARI TOSHIBA

## 6100 Darmstadt

**Heim**
**Büro- und Computermarkt**
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon (0 61 51) 5 60 57

## 6240 Königstein

**KFC**
**COMPUTERSYSTEME**
**Wiesenstraße 18**
**6240 Königstein**
**Tel. 0 61 74 - 30 33**
**Mail-Box 0 61 74 - 53 55**

## 6300 Gießen




## 6330 Wetzlar



Fachmarkt
für
Computer u.
Unterhaltungs-
electronic in Wetzlar,
**Einkaufszentrum Bahnhofstraße, Tel. (0 64 41) 4 85 66**

## 6457 Maintal

**Landolt-Computer**
Beratung · Service · Verkauf · Leasing

Wingertstr. 112
6457 Maintal/Dörnigheim
Telefon 0 61 81 - 4 52 93

## 6500 Mainz

**:ELPHOTEC**
Computer Systeme
**Ihr Atari Systemhändler**
**mit eigenem Service-Center**
Walpodenstraße 10
6500 Mainz
Telefon 0 61 31 - 23 19 47

## 6520 Worms

**ORION**
Computersysteme GmbH
Friedrichstraße 22
6520 WORMS
Tel. 0 62 41 / 67 57 - 67 58

## 6700 Ludwigshafen

*MKV Computermarkt*

*Bismarck-Zentrum*
*6700 Ludwigshafen*
*Telefon 06 21 - 52 55 96*

## 6720 Speyer

*MKV Computermarkt*
*Gilgenstraße 4*
*6720 Speyer*
*Telefon 0 62 32 - 7 72 16*

## 6800 Mannheim




Computersysteme + Textsysteme
**6800 Mannheim 24**
Casterfeldstraße 74-76
☎ (06 21) 85 00 40 · Teletex 6 211 912

**Computer-Center**
**am Hauptbahnhof GmbH**

**L 14, 16-17**
**6800 Mannheim 1**
**Tel. (06 21) 2 09 83 / 84**

## 6900 Heidelberg

**JACOM COMPUTERWELT**
**Hardware · Software**
**Schulung · Service**
**Mönchhofstraße 3 · 6900 Heidelberg**
**Telefon 0 62 21 / 41 05 14 - 550**

**HEIDELBERGER**
**COMPUTER CENTER**
**Bahnhofstraße 1**
**6900 Heidelberg**
**Telefon 06221/27132**

## 7022 L-Echterdingen

Autorisierter ATARI-
System-Fachhändler
**ATARI ST**




**Matrai Computer GmbH**
Bernhauser Str 6
7022 L-Echterdingen
☎ (07 11) 79 70 49

## 7030 Böblingen

**Verkauf – Service – Software**

Norbert Hlawinka
Sindelfinger Allee 1
7030 Böblingen
Tel. 0 70 31 / 22 60 15




**COMPUTER CENTER**

## 7100 Heilbronn

Unser Wissen ist **Ihr** Vorteil

**Walliser & Co.**
Mönchseestraße 99
7100 Heilbronn
Telefon (071 31) 6 00 48

## 7100 Heilbronn




**Computer-Welt**
Am Wollhaus 6
7100 Heilbronn
Tel. 0 71 31 - 6 84 01 - 02

## 7101 Flein

der **COMPUTERLADEN** von




**In der Falterstraße**
**7101 Flein**
Beratung, Service, Verkauf,
Software-Entwicklung
direkt beim Peripherie-Hersteller

## 7150 Backnang

**Computer-Fans finden bei uns alles von:**




## 7400 Tübingen

**Werner Brock**
**COMPUTERSTUDIO**
Poststraße 2—4 · D-7400 Tübingen
Tel. (070 71) 3 43 48 · Fax (070 71) 3 47 92
Autorisierter Systemfachhändler für:
**ATARI, Schneider, Commodore, Panasonic, Kaypro, Sharp, NEC, OKI, STAR...**

## 7410 Reutlingen

**Werner Brock**
**COMPUTERSHOP**
Federnseestr. 17 · 7410 Reutlingen
Tel. 0 71 21 - 3 42 87
Tx 1 72 414 024 RMI D · box:rmi:taisoft · Fax 0 71 21 - 33 97 79
Autorisierter Systemfachhändler für:
**ATARI, Schneider, Commodore, Panasonic, Kaypro, Sharp, NEC, OKI, STAR,...**

## 7450 Hechtingen




Gesellschaft für Datenverarbeitung mbh
Computer · Drucker
Zubehör · Fachliteratur
Schloßplatz 3 · 7450 Hechingen
Telefon 0 74 71 / 1 45 07

## 7475 Meßstetten

**Ihr ATARI-Systemhändler im Zollern-Alb-Kreis**
HEIM + PC-COMPUTERMARKT
**HARDWARE · SOFTWARE · LITERATUR**




ATARI COMMODORE CUMANA DATA-BECKER
MULTITECH RITEMAN SCHNEIDER THOMSON
7475 Meßstetten 1 · Hauptstraße 10 · 0 74 31 / 6 12 80

## 7480 Sigmaringen

Wir stellen uns für Sie auf den Kopf!

Wir führen:
Hardware - Software - Zubehör
Zeitschriften - Bücher

Wir sind autorisierte Fachhändler von:
ATARI Schneider
Tandon

SOFT&EASY COMPUTER GMBH
7480 Sigmaringen, Am Rappgassle, Tel. 07571/12483
7968 Saulgau, Pfarrstraße 13, Tel. 07581/2598

## 7500 Karlsruhe

ERHARDT Am Ludwigsplatz
Am Ludwigsplatz · 7500 Karlsruhe 1 · Tel. (0721) 1608-0

MKV GMBH
Kriegsstraße 77
7500 Karlsruhe
Telefon (0721) 84613

## 7600 Offenburg

FRANK LEONHARDT ELECTRONIC
Ihr Fachgeschäft für Microcomputer · Hifi · Funk
In der Jeuch 3
7600 Offenburg
Telefon 0781/57974

## 7640 Kehl/Rhein




Badstrasse 12
Tel. 07807/822
Telex: 752 913
7607 NEURIED 2

Filiale:
Hauptstrasse 44
Tel. 07851/1822
7640 KEHL/RHEIN

eigener Service · eigene Software
ELEKTRO-MÜNTZER GmbH

## 7700 Singen

U. MEIER
Computersysteme
7700 Singen-Htwl.
Am Posthalterswäldle 8
Telefon 07731-44211

## 7730 VS-Schwenningen

BUS BRAUCH & SAUTER COMPUTER TECHNIK
Villinger Straße 85
7730 VS-Schwenningen
Telefon 07720/38071-72

## 7750 Konstanz

ATARI ★ PC's ★ SCHNEIDER
computer-fachgeschäft
rösler
Rheingutstr. 1 · ☎ 07531-21832

## 7800 Freiburg

CDS EDV-Service GmbH
Windausstraße 2
7800 Freiburg
Tel. 0761-81047

PYRAMID COMPUTER GMBH
KARTÄUSERSTRASSE 59
D-7800 FREIBURG/BRST.
TELEFON 0761-382038

## 7890 Waldshut-Tiengen



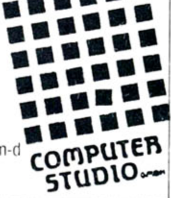

service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 07751/3094

## 7900 Ulm

HARD AND SOFT COMPUTER GMBH
Ulms großes Fachgeschäft
für BTX, Heim- u. Personalcomputer
Herrenkellergasse 16 · 7900 Ulm/Donau
Telefon 0731/62699

EDV-Systeme
Software-erstellung
Schulung

Systemhaus:
Frauenstraße 28
7900 Ulm/Donau
Tel. (0731) 28076
Telex 712973 csulm-d




## 7918 Illertissen

biotech gmbh
technische Informationsysteme
Computerladen
Marktplatz 13
7918 Illertissen
07303/5045

## 7980 Ravensburg

GRAHLE
Expert Grahle Computer
Eisenbahnstr. 33
7980 Ravensburg
Tel.: 0751/15955
Vertragshändler für ATARI, Schneider und Star

## 8000 München

star NEC



HDS-COMPUTER-VERTRIEBS GMBH
Am Klostergarten 1 · 8000 München 60
Tel. (089) 837063-64
ATARI Commodore

ludwig
COMPUTER + BÜROTECHNIK
COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTÄDTER STRASSE 62L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089/3113066 · TELETEX 898341

schulz computer
Schillerstraße 22
8000 München 2
Telefon (089) 597339
Beratung · Verkauf · Kundendienst

## 8032 Gräfelfing




COMPUTER SYSTEME SCHULUNG
Am Haag 5
8032 Gräfelfing
Tel. 089-8545464, 851043

## 8100 Garmisch-Partenk.

Uwe Langheinrich Elektronik Center
Hindenburgerstr. 45
8100 Garmisch-Partenkirchen
Tel. 08821-71555
Bitte Gratisliste anfordern

## 8150 Holzkirchen

**ATARI**

Besuchen Sie uns!

Fordern Sie unseren Software-Katalog (520ST) an!



**MÜNZENLOHER GMBH**
Tölzer Straße 5
D-8150 Holzkirchen
Telefon: (0 80 24) 18 14

## 8170 Bad Tölz

**Uwe Langheinrich**
**Elektronik Center**
Wachterstr. 3
8170 Bad Tölz
Tel. 0 80 41 - 4 15 65
Bitte Gratisliste anfordern

## 8330 Eggenfelden




Computer-Centrum
R. Lanfermann

Schellenbruckstraße 6
8330 Eggenfelden
Telefon 0 87 21 65 73

Altöttinger Straße 2
8265 Neuötting
Telefon 0 86 71 7 16 10

## 8400 Regensburg

**Zimmermann**
**elektroland**

8400 Regensburg
Dr.-Gessler-Str. 8
☎ 09 41 / 9 50 85

8390 Passau
Kohlbruck 2a
☎ 08 51 / 5 20 07

## 8423 Abensberg

**COMPUTERVERSAND**
**WITTICH**
Tulpenstr. 16 · 8423 Abensberg
☎ 0 94 43 / 453



## 8500 Nürnberg

**EINE IDEE ANDERS**
KARSTADT NÜRNBERG AN DER LORENZKIRCHE




1. KLASSE EINKAUFEN IM WELTSTADTHAUS

**hib**
HIB Computer GmbH
Äußere Bayreuther Str. 57a – 59
8500 Nürnberg 10
Tel.: (09 11) 56 29 26 · Telex: 17 – 91 18 253
Teletex: 2627 – 91 18 253 · Telefax: (09 11) 51 30 40
Systemfachhändler für anspruchsvolle Computertechnik

Microsoft · TOSHIBA · SONY · ATARI · brother · Zenith data systems · EPSON

## 8520 Erlangen




loewenichstr. 30 - d - 8520 erlangen
telefon 09131 / 2 50 18
telex 62 97 65 atron d

**Computerservice**
**Decker**
Meisenweg 29 - 8520 Erlangen
Telefon 0 91 31 / 4 20 76

**Zimmermann**
**elektroland**

8520 Erlangen
Nürnberger Straße 88
Tel. (0 91 31) 3 45 68

8500 Nürnberg
Hauptmarkt 17
Tel. (09 11) 2 07 98

## 8600 Bamberg




Bamberg · Tel. 0951 / 2 78 08 - 09

## 8700 Würzburg

**SCHOLL**
**BÜROTEAM**

Hardware · Software
Service · Schulung

**computer center**
am Dominikanerplatz
Ruf (09 31) 3 08 08 - 0

## 8720 Schweinfurt

*Uhlenhuth GmbH*
Computer + Unterhaltungselektronik
Albrecht-Dürer-Platz 2
8720 Schweinfurt
Telefon 0 97 21 / 65 21 54

## 8900 Augsburg




**Unser Plus: Beratung u. Service**
Schwalbenstr. 1 · 8900 Augsburg-Pfersee
Telefon (08 21) 52 85 33 oder 52 80 87
Computer Vertriebs- und Software GmbH

**Bei uns werben bringt**

**GEWINN**



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057

BUF

## 8940 Memmingen

**EDV-Organisation**
**Hard- + Software**
**Manfred Schweizer KG**
Ulmer Str. 2, Tel. 0 83 31 / 1 22 20
8940 Memmingen

## ÖSTERREICH

## A-1020 Wien

**W + H.**
**Computerhandel Ges m.b.H.**

Förstergasse 6/3/2 · 1020 Wien
Tel. 02 22 - 35 09 68

## A-1040 Wien

**Ihr ST-Fachhändler in Wien**
**Computer-Studio**
Wehsner Gesellschaft m.b.H.

1040 Wien - Paniglgasse 18-20
Telefon 02 22 - 65 78 08, 65 88 93

## A-8010 Graz




A-8010 GRAZ
Mandellstraße 23;
Tel.: (0316) / 70 28 40-0*, 70 28 93-0*
Tx.: 31 25 34 zupan a

# SCHWEIZ

## CH-1205 Geneve

PIMENT ROUGE
INFORMATIQUE S.A.
8, RUE DES MARAICHERS
1205 GENEVE TEL. 022/28 56 24

## CH-1700 Fribourg

FRIDAT SA INFORMATIQUE
ehem. Softy Hard's Computershop
VOTRE SPECIALISTE
Rte des Grives 4
1700 Granges-Paccot/Fribourg
Tel. 0041 (0)37 26 66 28
Fax. 0041 (0)37 26 61 06

## CH-2503 Biel

*UE*
***URWA Electronic***
Ihr ATARI ST Spezialist in der Schweiz.
0 32 / 25 45 53
Lindenweg 24, 2503 Biel

## CH-3084 Bern

Computer Corner
Ihr ATARI ST-Partner in Bern
Hard-Software  Beratung
Midi-Schulung  Service
Seftigenstr. 240 3084 Bern-Wabern
Zentrum Wabern Tel. 031/54 51 00

## CH-4054 Basel

**Wir sind Amiga- und ST-Freaks.**
Unsere Öffnungszeiten:
Von Dienstag bis Freitag
9.30 – 12.30 und 14.00 – 18.30 und
am Samstag 9.30 – 16.00




Basel: Tel. 061/39 25 25 · Holeestrasse 87 · 4054 Basel
Aarau: Tel. 064/22 63 33 · Kasernenstrasse 26 · 5000 Aarau

## CH-4625 Oberbuchsiten

**STECTRONIC M. Steck**
*Electronic-Computer-Shop*
*Hauptstr. 104/137*
*CH-4625 OBERBUCHSITEN*
*Tel. 0 62/ 63 17 27 + 63 10 27*

## CH-5430 Wettingen



Senn Computer AG
Zentralstrasse 93 Tel. 056 / 27 16 60
CH-5430 Wettingen Telex 814 193 seco

## CH-8001 Zürich

ADAG LASERLADEN
SEILERGRABEN 41 . 8001 ZÜRICH
MO-FR: 12-18.30 TEL. 01/251 49 34
DTP... LASERPRINTS ... ET CETERA!

## CH-8006 Zürich

ADAG
COMPUTER-SHOP
UNIVERSITÄTSSTR. 25
8006 ZÜRICH TEL. 01/252 18 68

**Computer-Center P. Fisch**
Stampfenbachplatz 4
**8006 ZÜRICH**
01 / 363 67 67

## CH-8021 Zürich

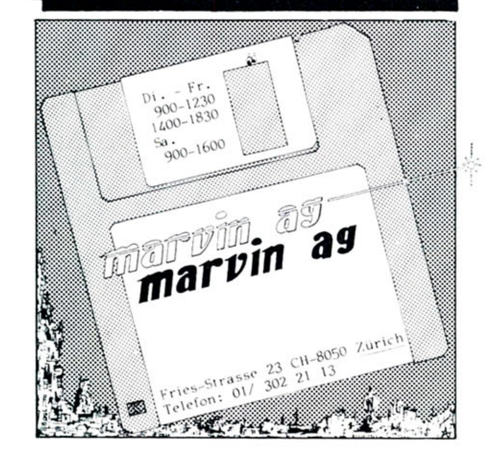



Das Warenhaus der neuen Ideen
Ihr Computer-Fachhandel an der
Bahnhofstrasse 75 · Zürich

## CH-8050 Zürich



Senn Computer AG
Langstrasse 31
Postfach Tel. 01 / 241 73 73
CH-8021 Zürich Telex 814 193 seco




## CH-8052 Zürich




zentral in Seebach, Schaffhauser Str. 473, 8052 Zürich
Telefon: 01-302 26 00
Versandkatalog anfordern

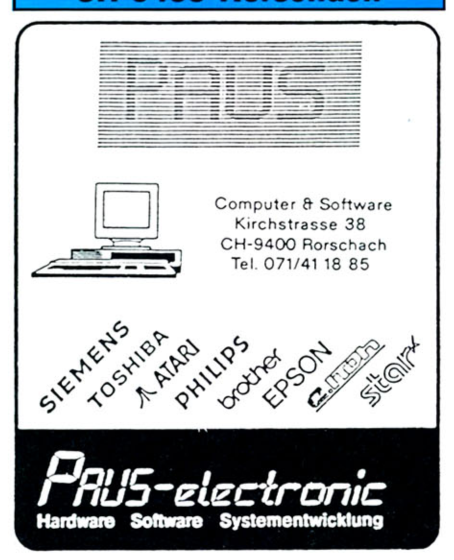
## CH-8200 Schaffhausen

ZIMELEC
CAR-AUDIO
COMPUTER + ELECTRONIC
Bachstraße 28 · 8200 Schaffhausen
Tel.: 053/55224
Montag—Freitag 9.00—12.00, 13.30—18.30
Samstag 9.00—16.00

## CH-9000 St. Gallen

ADAG
COMPUTER-SHOP
TORSTR. 25
9001 ST.GALLEN
TEL. 071/25 43 42

## CH-9400 Rorschach




## LUXEMBURG



# Kleinanzeigen

## BIETE HARDWARE

●●● TOP-ANGEBOT ●●●
Brenne TOS + BlitterTOS – billig! Auch beide Systeme zusammen im ST! **Kinderleichter** Einbau! Gratisinfo: M. Meyer / G. Rohlfs-Str. 54c / 282 Bremen 70

---

**ST-Harddisk 40 MB** anschlußfertig incl. Pilzfuß für Monitor 1850 DM Tel. 02371/60951

---

Wegen Systemaufgabe günstig zu verkaufen, Hardware, Software und Literatur. Josef Hammer Fichtenstr. 11 · 8551 Röttenbach

---

Vortex HDplus20, neu, unbenutzt, originalverpackt, umständehalber abzugeben – 07136/22297 –

---

**Drucker Epson FX80** VB 500,– Sperber, Tel. 0951/58028 abends

---

260 ST 1/2J. 250,-/260 ST 1MB 350,- Okimate20 Farbdruck. f. Atari 1/2J. 200,- / SF354 o. Netzg. 100,- / BASF6138 5¼"/DS/80Tr/3ms/ Shug. m. Manual 200,-/Orig. Maus unbenutzt 80,- / 260ST Gehäuse + Tastatur unben. 50,-
Tel. 089/6091877 ab 17.00 Uhr

---

1040 STF + Mon. + Mouse VB 1280 DM
Guido Henn, Tel. 02257/7940

---

Omikron-Basic Modul-Vers. + ausführl. Literatur DM 150 kompl. Tel. ab 18 Uhr 02680/8929

---

**Blitter-ROM-TOS 95 DM**
**Tel. 02630/7525**

---

★ ★ ★ ST-LASER-SERVICE ★ ★ ★
★ Ausdruck von **Studienarbeiten,** ★
★ **Diplomarbeiten** und anderen ★
★ **anspruchsvollen Schriftsätzen,** ★
★ Einladungen, Serienbriefe usw. ★
★ in **höchster Qualität** auf einem ★
★ **Laserdrucker!** Auch d. Ausdruck ★
★ von Grafiken, Werbung, Visit. ★
★ od. ähnlichem möglich. Kosten- ★
★ lose Info anfordern! Tel. Nr. ★
★ 02323/46362, Btx: 0232346329 ★
★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

---

**HF-Mod. und Trafo,** 1a Zustand, 120,– DM 07242-6652

---

**Wordplus, Beckertext, Starwriter ST-Drucktreiber** f. **alle Star NL-10** u. **LC-10** + Col. Kompl. angep. volle Befehlsausnutz.; 1.5.zeil, je 30,–; **DIC 241 K** 10 DM; U. Köhler, Mont-Cenis Str. 537, 4690 Herne 1

---

**SCANNER** (Fertigger.) nach Heft 4/88 incl. PD-Disk V1.2 DM 92,– Tel. 06105/24527 ab 19.00 Uhr

---

**SM124 Multimode** nach ST-Comp. Komplettumrüstung DM 160,– Tel. 06105/24527 ab 19.00 Uhr

---

**Festplatten für Atari & Amiga**
Atari: 260ST – Mega-4, Amiga 1000/2000. Anschlußfertig im Gehäuse mit Treibersoftware (nach CT') z. B. 20 MB netto für DM 850 größere Kapazitäten verfügbar
RCE-GmbH
Gleißbühlstr. 7, 8500 Nürnberg 1
Tel. 0911/2435 04/14

---

**SM124 + SF314 je 280, Maus 50, Platine + Tastatur für Bastler 150, ROMport Leerplatine (4) 20 Omicron Int. V2.06 089/9828497**

---

2MB Ram, steckbar, von Weide: VB 1000 DM.
M. Wolf, Tel. 0641-41450

---

Original **Atari Harddisk SH205** 3 Monate jung, mit Garantie! Oliver Meissel, 0761/702497, VB

---

**SF 354** fast neu in Originalpack 150 DM 07227-3099 (ab 19 Uhr)

---

**RHO-Buserw.** Atari 1040STF, P1O-Board, PC-Geh. **Neu** DM 2.950 Tel. 0228/616606

---

9-Nadeldrucker **OKI 192+,** ST-V. 1 Jahr, DM 550 Tel. 06142/22677

---

Verk. Diskstad.354: DM 130,– Tel. 06221-13443 · Div. Orig. Software

---

**XXX Verkaufe XXX**
Floppy SF 354 150,- DM
CSF-Gehäuse 100,– DM
SUCHE: Festplatte SH204.
Tel.: 09392/7874

---

SF 354 99,– DATAPHON S 21d, Tel. 06421-84675

## BIETE SOFTWARE

MODULA-2 Compiler; 07386/588

---

●●● TOP-ANGEBOT ●●●
1. SUPER PD-Software...
2. Lernsystem, GEM, viele Modi
3. GfA-Basic Shell 4. Speedchip
Gratisinfo: M. Meyer / G. Rohlfs-Str. 54c / 282 Bremen 70

---

Digitalis. mit **TURBO DIZER?**
Endlich **komfortable Software!**
Slider-Helligkeitsregler
10 Bildspeicher, Animation
Autom. Belichtungsanp. ★ **30,–**
A. Hill, Schedestr. 4, 5300 Bonn

---

■ Verkaufe Originalprogramm
**LOGISTIX**
DM 295,– N. Fischer, Hobacherweg 6, CH-6010 Kriens
Tel: 0041/41/45 83 69

---

**PD-MB:** 0571/710141 ★ 300/8/N/1

---

★ ★ Public Domain Software ★ ★
IBM u. komp. Disk ab 6,– DM
Atari ST Disk ab 3,– DM
Riesenauswahl ★ ★ ★ Superpreise
Katalog-Disk für je 1,80 DM
Graf & Schick · Hauptstraße 32 a
8542 Roth · Tel. 09171/50 58-59

---

Lohn-Einkommensteuer, Miet-Lastenzuschuß, Rentenber./Beamtenversorgung.
H-I-SOFTWARE
Niederfelderstraße 44
8072 Manching · Tel. 08459-1669

---

PD-Grafik-Bilder, Art-Lib für Degas + Stad. Katalog: Frey, Rheinstr. 12A · 6538 Münster-Sarmsheim

---

**Von Lehrer für Lehrer!** Notenverw.prg **Markbook plus** in GEM. Über 20 versch. Funkt. f. alle gäng. Notensyst. Spez. vers. auf Wunsch; Info gegen Freiumschlag.
H. J. Merkel/Nahestr. 28/6600 SBR

---

Professionelle **LOTTOSYSTEME**
Info S. Richter, Rilkestr. 8, 4445 Neuenkirchen, Tel. 05973/5157

---

■ **Fußball-Bundesliga** für alle ST.
■ Alle Ergebnisse, Tabellen und
■ Statistiken seit 1964! Leicht zu
■ aktualisieren. Für jeden Drucker.
■ Mausbedienung. Gratisinfo an-
■ fordern oder gleich bestellen.
■ 40,– DM Bar, Scheck + NN.
■ Andreas Smoor, Tannenstr. 50,
■ 4460 Nordhorn

---

**IsGemDa** 2.0 110,– M. Vetter L.-Jahnstr. 17 · 7835 Teningen 1 07641/459277 (bis 17.00 Uhr

---

**PD-Software** ab 2,– DM/Diskette Angebot wie PD dieser Ausgabe Tel. 02721/2432

---

■ ■ PD-SERVICE ■ ■
■ PD 1 bis PD dieser Ausgabe ■
■ Einzeldiskette SS **DM 4,00** ■
■ Doppeldiskette DS **DM 6,00** ■
■ beliebig kombinierbar) ■
■ **ab DM 50,00 10 % Rabatt** ■
■ Porto u. Verpack. DM 3,00 ■
■ V.-Scheck o. NN (+DM 3,50) ■
■ N. Twardoch, Gröchteweg 22 ■
■ 4902 Bad Salzuflen 1 ■

---

★ ★ ST-PD, Kopie ab 2,50 ★ ★
alle PD aus ST-Comp. u. eigene Info: T. Helfers, Portslagerstr. 30, 2905 Edewecht, T.: 04405/6809

---

**Restposten: Vocmaster V: 2.3**
Superlatives Vokabelprogramm, Synonyms, GEM, Menüleiste,
**Sonderpreis 39,– Info gratis**
Reiner Kocher T: 089/3134946
Caracciolastr. 16, 8 München 45

---

**Lohnsteuerjahresausgleich und Einkommensteuer 1987/88**
**neu: Berlin-Präferenz / 30 DM**
Jochen Höfer, Grunewald 2a, 5272 Wipperfürth 02192/3368

---

★ **EUSAX** mit PD-Aktuell dem Infomagazin. Fordern Sie unser Gratisinfo an. ★ **NEU 24-h-Best. und Liefer-Service!!** EUSAX PD **Augustastr. 3** 4018 Langenfeld (02173/81232) **Rund** um die **UHR!**

---

**PLOT_ST für FORTRAN**
Jetzt auch für AC-FORTRAN! Einfachste Ausgabe grafischer Darstellungen auf Bildschirm und Drucker. Mehrere Funktionen gleichzeitig darstellbar. Kompatibel mit Großrechner Standard (Plot-79) DM 95,–. Handbuch DM 15,– (wird verrechnet); Info gratis.
**M. Gamer,** Friedrichsring 26, 6050 Offenbach

---

Lattice C 3.04 (deutsch) 200,– SF 354 120,– 08221/4488

---

■■PD-Versand in Rhein-Main■■
■ Alle ST-PD's für nur 6 DM ■
■ Copy-Service 5¼ Z auf 3½ Z. ■
■ nur 9 DM (Markendisketten) ■
■ Digital Image, Pf. 1206, 6096 ■
■ Raunheim, 06142/22636 ab 19⁰⁰ ■

---

STEVE 3.0 280 DM, Steinberg 24 + Masterscore 680 DM, Acopy kopiert alles 60 DM, viele Spiele, Tel. 02257/7940

---

**ST-PD Disketten** dieser Ausgabe sowie Aladin/MS-DOS PD **ab 2,– Info:** A. Witaseck, Josef-Neuberger Str. 26, 4000 Düsseldorf

---

Blitter-TOS + altes TOS gleichz. umschaltbar f. alle ST, Anl.
K. Ratsch, Herner 127, 4350 Recklh 02361/28442

---

**OMIKRON** - Entwicklungssystem Basic + Compiler NP 360,– für nur 270,– Tel: 07433/34498

---

Diverse SW-Prg z. B. VIP-Prof. Logistix usw. Liste kostenlos Info nach 18 Uhr Tel. 073086461

---

**Noch ein Etikettendrucker für 3,5"-ST-Disketten! Aber dieses Mal der Beste.** bSE druckt klar und übersichtlich alle Dateinamen und sonstige notwendige Informationen (Kurzbeschreibung, Disknr. usw.), die sich auf der Diskette befinden, **ohne überflüssigen Firlefanz!!!** DM 39,– + Versandkosten von DM 5,– bei Scheck oder DM 10,– bei NN. M. Kremers, An der Kreuzstr. 62, 4057 Brüggen 1, Telefon: 02163/59711

---

**Wärmebedarfsberechnung DIN4701**
**KZahlberechnung DIN4108**
**Dampfdiffusion-Tauwasserschutz**
**Wärmeschutznachweis WäSchV**
Dipl. Ing. V. Koch, Am Mehnacker 11 3563 Dautphetal 3, Tel. 064687652

---

Layoutprogramm für Atari ST für DM 69,– (Info 1,–)
Reiner Rosin, Peter-Spahn-Str. 4, 6227 Oestrich-Winkel

---

**BECKERPAGE ST** neu, nicht installiert DM 299 Tel. 07351/6528

---

**Public-Domain-Blitzversand!**
**Riesenauswahl! Preise:**
**ab DM 4,– einseitige Disketten**
**ab DM 6,– doppelseitige**
**Inclusive Diskette!!!**
**Auch alle ST-Disks! Auf Doppeldisks beliebig kombinierbar!**
**Gratisliste anfordern bei:**
**A. Gauger Software**
**Buhlstraße 16a, 7505 Ettlingen**
**0 72 43 / 3 18 28**
**Bitte Computertyp angeben!!**

---

■■■**Public Doman 5,50 DM!**■
■ PD aller ST-Ausgaben auf ■
■ **Markendisks SS je DM 5,50** ■
■ P&V. DM 5,00, ab 20 St. frei! ■
■■ **Achtung!!** ■■ **Achtung!!** ■■
■ **Neue PD auch im „ST-Comp."** ■
■ **„Sommerloch"!!!** – ART-Lib's ■
■ für Degas, Stad, Sig., Word+, ■
■ SIGNUM Fonts/Util. u.v.m.!! ■
■ WO? – **Schauen Sie** ■
■ **nach der Anzeige an anderer** ■
■ **Stelle in dieser Ausgabe!** ■
■ **NEUEN Katalog bestellen!** ■
■ (für 5,– incl. Probedisk!!) ■
■■■**PD-Express-Versand**■■■■

---

**ORIGINALPROGRAMME**
**Riesenauswahl an neuesten Programmen zu den günstigen –Gauger Software Preisen–**
**... die Gelegenheit!!**
**Gratisliste anfordern bei:**
**A. Gauger Software, Buhlstr. 16a 7505 Ettlingen, 07243/31828**

---

**Datenbank IsGemDa V2.0** mit Schnittstelle zu C/Modula/GFA-Omikron-Basic/Pascal
**Original für 150,– DM**
Tel: 09521/69324 8–17 Uhr.

---

GEM-Baustatik-Software! Komf. Bedienerführung/Grafikausgabe Info o. Demo-Disk (10 DM bar/VS) von: G&S-Soft, Jakobstr. 4, 6100 Darmstadt, 06151 591009/711156

---

■■ **GFA-SHELL incl. Linker!** ■
Neue Befehle: **Import, Export, ...** Dialogbox-/Fensterverwaltung... Mit Anleitung für **nur 30,– DM!!** O. Krahmer, Potsdamer- 22, 28HB1

---

PD-Grafik, Art-Lib für Degas + Stad. Katalob: Frey, Rheinstr. 12A 6538 Münster-Sarmsheim

---

TDI Modula-2 Development 250,-
Wordstar 3.0/Mailmerge... 100,-
Borrowed Time .......... 30,-
Mindshadow ............ 35,-
alles Originale! (0621) 311271

---

Verk–Tausche Spiele Orig. Karat **bes. TDI-Quellmodule** (auch Kauf) Tel. 07144/14559 (Joachim)

---

**ANTI-VIRUS V1.5**
Gratisinfo von: M. Klocke-Sewing Gronauer Str. 19 · 4800 Bielefeld

---

**V-Manager Prosy** 1.1 f. Atari ST **Das** Versicherungs-Agentur-Prg. NP 1500,– DM für 990,– DM wg. umstieg auf Firmenrechner zu verk. Demo vorhanden.
Tel. 030/3961032

---

Weg mit dem Streß! Schneller planen mit **PROJEKT.** Prof. Netzplanprog – einfache Bed. 348 DM INFO: HS-Soft, Wachbacher Str. 5 6990 Bad Mergentheim

★ Achtung Easy Draw Freunde ★ Das Programm Easy Konvert konvertiert alle Bilder die mit dem Screenformat 32 K z. B.: Mono-STar, Doodle erstellt worden sind, so das Sie später mit dem Programm Easy Draw geladen werden können. PREIS 39,90 DM. Adr.: Thomas Berdowski, Postf. 1807, 6200 Wiesbaden, Tel. 06121-301102/374314

**Hndschriftenanalyse + Vergleich** 180 Merkmale. Disk, Buch DM 38 M. F. Feldstedter 6, 1000 Bln 49

**Baustatik-Software** f. ATARI ST **Ebene** und **räumliche Fachwerke** mit graph. Darstellung. Info gegen Rückporto (DM 0,80) Dipl. Ing. B. Gerbig; Kirchenweg 3 6314 Ulrichstein 1

Original-Software: Campus V1.2 700 DM; 1st-Word+ 300 DM, GfA-Basic V 2.0 120 DM, GfA-Comp. 130 DM, ViP-Prof. 300 DM. 030/7828245

**PD-Software** ab 2,– DM/Diskette von PD 1 – PD dieser Ausgabe Tel: 02721/2432

★★ FIBU-PRG DM 59,– ★★ GEM unterstützt B/D-Ausgaben 60 Konten / 2500 Buchungen monatlich E/Ü-Rechn GuV/Bilanz MwSt autom. Tel. 09429/8402

■■PD aus „ST" nur 4,80 DM■■ ■ incl. Markendisk. Ab Nr. 1 ■ ■ 24-h-Bestellservice. ■ ■■ Schneider 0281/61772 ■■

org. **Eplanpro** mit Anleitung 99,– **Address-Access** m. A. 25,– DM Info bei Schistek Andreas 8510 Fürth, Schwabacher Str. 260

WELTNEUHEIT für ATARI-ST Video-Film im Computer Software für Erwachsene Fischer; Pf. 1619; 4830 Gütersloh

Public-Domain ★ Gratisinfo ★ Neue Software ★ Super Preise ★ Tlf. 02954-1050 ★ M. Schönfelder Südstr. 16, 4784 Rüthen 5

★ **Börsenprg Depotdeluxe** + Kurse f. 300 Aktien! 320 DM, **Logistix** 300 DM; 09802/7773 nach 17 Uhr

**Wärmebedarf DIN4701 + K-Zahl** ★ Heizflächenauslegung ★ Rohrnetz ★ Demodisk 2-seitig für DM 10 Vorkasse von J. Binder, Eichendorffstr. 15 · 5030 Hürth

## SUCHE SOFTWARE

Für **Aladin** System 2.0d Pagemaker 1.2 05764/1213 M. Föllinger

Suche Volltextdatenbankprogramm für ST unter TOS oder and. impl. Betriebssystem – Dr. K. Gabrisch, U 6, 11; 6800 Mannheim 1 Tel. 0621/25274

Suche Signum2, STAD, Vocroyal, 1st Wordplus 2.02, Campus Draft, K-Graph2, Easydraw 0228/626526

## SUCHE HARDWARE

★★★ DEFEKTER ST? ★★★ Biete 50 – 100 DM Tel. 07245/2168

## TAUSCH

**Neu!! PD-Software zum Nulltarif – keine Kopierkosten!!** Tausch gegen Leerdisks oder neue PRGe.! Für Liste: Disk & Rückporto an: D. Metz; Bahnhofstr. 84A; 2153 Neu Wulmstorf

Ich digitalisier Eure Fotos Tausche auch PD-Soft (Grafik usw.) Tel. bei Frank 05241/28015

KONTAKTE

MAINSTREAM – Der User-Club für Atari-ST-Anwender Info gegen Rückporto von: Kay-U. Berghof Roseggerstr. 5 5600 Wuppertal 2

LB Computer sucht PROGRAMMIERER, die in Megamax Modula 2 programmieren wollen. Folgende Themen sind geplant: Textverarbeitung, Dateiverarbeitung, Kalkulation, Geschäftsgrafik, Modemprogramm, Btx Programm, Tools, und andere nach Wunsch. LB Computer Pf 1253 · 8425 Neustadt/Donau

## VERSCHIEDENES

Computer + Electronic-Börse: 4.+5.6. 4250 Bottrop Saalbau, 11.+12.6. 4200 Oberhausen Saal am Revierpark, 17.–19.6. 4010 Hilden Stadthalle, 25.+26.6. Düsseldorf Rheinterassensaal. Jeder kann teilnehmen. Tel. 02845-27260

(VER)KAUF: Hard + Soft: GG-Versand Postfach 1450 8312 Dingolfing

**Überspiele 5,25" > 3,5"**; CP/M > TOS; CP/M > MDOS; MDOS > TOS; ab 5,–/Disk konvert. Wordst > STWord + Signum2 ab 1,–/Datei. A. Zimmermann, Küchenmeisterstr. 5, 8711 Rödelsee

Computer + Elektronik-Börse: 2.+3.7. 5170 Jülich Stadthalle, 9.+10.7. 4040 Neuss Nordstadthalle, 16.+17.7. 4220 Dinslaken Saal Hackfort, 17.7. Köln Hauptbahnhof, 23.+24.7. 4019 Monheim Festhalle, 30.+31.7. 4130 Moers Motel Moers. Jeder kann teilnehmen. Info: 02845-27260 abends

**ST-Computer,** Jahrgänge 86/87 Komplett 100 DM, Tel. 08167/8671

Übernehme Programmieraufgaben in GFA-BASIC P. Neurohr Eschersheimer Landstr. 273 6 Frankfurt 1 Tel: 069/567298

# FÜR IMMER UND EWIG...

**muß es ja nicht gerade sein, allerdings möchte man doch ab und zu mal ein Programm im Computer starten, das nach Verlassen der Initialisierung weiterhin vorhanden ist. Solche Vorgehensweisen benötigt man meist dann, wenn eine Routine im Interrupt laufen soll. Das diese Programminitialisierung unseren Lesern Probleme macht, erkennt man an den vielen Zuschriften, in denen danach gefragt wird.**

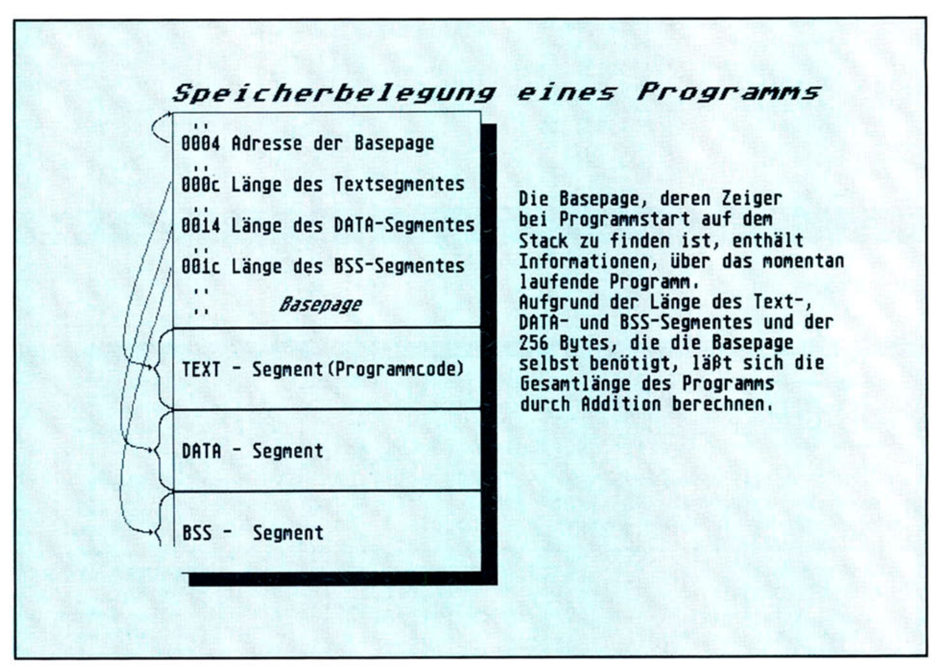



*Bild 1: So sieht ein Programm im Speicher aus*

In der letzten Ausgabe haben wir uns mit der Heilung des Blitters beschäftigt - wie geht es Ihrem Blitter, gut? - und dabei eine Routine in das Betriebssystem eingebunden, die den Busfehler abfragt. Nach einer kurzen Initialisierung wurde die eigentliche Routine im Speicher RESIDENT verankert. Es heißt übrigens tatsächlich resident und nicht (!) resistent, wie viele meinen, denn resident heißt frei übersetzt ortsansässig, während resistent widerstandsfähig bedeutet. Da wir unser Programm aber nicht vor irgendwelchen Viren schützen sondern nur im Speicher verankern wollen, ist resistent sicherlich nicht ganz passend. Trotzdem ist in der Zeit der Virus-Construction-Sets und böswilligen Hacker sicherlich ein Algorithmus angebracht, der so manche Programme und Disketten resistent macht - nur, gegen Viren ist man nicht gefeit und eine Betriebssystemroutine gibt es dafür schon gar nicht.

Zunächst wollen wir uns anschauen, wie ein Programm im Computer aussieht. Bei einem Rechner wie dem ATARI ST sind die Programme auf der Diskette nicht so gespeichert, wie sie später auch im Speicher vorliegen. Dies liegt erstens daran, daß ein Programm später an einer vorher nicht bestimmbaren Adresse liegen wird und zweitens durch große Variablenbereiche sehr viel Platz auf der Diskette verschwenden würde. Man kann also nicht sagen, daß eine Variable an einer bestimmten Adresse liegen wird, sondern man legt diese Variable an das Ende des für das Programm benötigten Speicherplatzes (TEXT-Segment) und adressiert dann relativ zum Befehl. Speichert man ein Programm mit solchen Variablen ab, weiß man aber nicht welche Adresse diese Variable haben wird. Das Auffüllen der Platzhalter mit Adressen, das (natürlich) vor dem Starten des Programms geschieht, nennt man Relozieren. Für diesen Reloziervorgang ist in der Programmdatei eine Tabelle vorhanden, anhand derer der Vorgang abgearbeitet wird.

Nun gibt es Variablen, die schon vordefiniert, also mit Anfangswerten belegt sind, und andere, die keinerlei Vorbelegung haben. Die zweite Art von Variablen wird häufig für diverse Puffer und sonstige große Speicherbereiche verwendet. Stellen Sie sich vor, Ihr Programm benötigte ein Megabyte Speicher, und Sie müßten diesen Bereich in ihrer Programmdatei miteinbauen. Spätestens dann sollten Sie sich überlegen, ob Sie Ihr Programm nicht gleich auf einer Harddisk ausliefern. Glücklicherweise müssen Sie das nicht, da es ein Segment in ihrem Programm gibt, das beschreibt, welche Variablen in welcher Größe benötigt werden. Erst beim Starten des Programms wird dann entsprechend den Angaben Speicherplatz reserviert und die Adressen im Programm eingetragen.

## Zusammengefaßt

Den Bereich der schon mit Werten vorbelegten Variablen nennt man DATA-Segment, BASIC-Fanatiker werden gewisse Ähnlichkeiten zu DATA-Zeilen erkennen, während der zweite Bereich als

BSS-Segment bezeichnet wird. BSS bedeutet Block-Storage-Segment (eigentlich ist das zweite s und das Wort Segment doppelt gemoppelt, aber was soll's, es hat sich so eingebürgert, also bleiben auch wir dabei), da in ihm die Angabe über das Organisieren größerer Blöcke im Speicher zu finden sind. Wir wissen also, daß unser Programm im Speicher aus einem TEXT-Segment, das nicht, wie man vielleicht denken könnte, die Ausgabetexte des Programms sondern den eigentlichen Programmcode enthält, einem DATA-Segment, welches vordefinierte Variablen- auch Programmtexte - beinhaltet und einem BSS-Segment besteht. Wenn das BSS-Segment im Speicher des Rechners vorliegt, verbraucht es natürlich den vordefinierten Speicherplatz der gewünschten Puffer (beispielsweise unsere oben erwähnten 1 Megabyte).

## Die Basis aller Dinge

Netterweise wird jedem Programm im Speicher von GEMDOS noch eine besondere Informations-Liste spendiert, die sich BASEPAGE nennt. Wie der Name schon sagt, liegt dieser Speicherbereich zu Beginn des Programms (BASE) und hat (wie es bei einer PAGE nun einmal üblich ist) 256 Bytes Länge. Wir wollen uns diese BASEPAGE einmal genauer anschauen.
Wir finden außer einigen momentan für uns nicht so wichtigen Informationen auch die Längen der einzelnen Segmente. Stellt sich die Frage, warum diese Längen überhaupt interessant sind! Dazu sollten Sie wissen, daß GEMDOS beim Laden des Programmes den größten, freien, zusammenhängenden Speicherplatz reserviert. Es ist dann Sache des Programmes, den benötigten Speicherplatz wieder schrumpfen zu lassen. Dies geschieht mit der GEMDOS-Routine Mshrink, die als Parameter den Beginn des Blocks und die Länge benötigt. Der Beginn unseres Blocks ist der Beginn der BASEPAGE, und die Länge ergibt sich aus der Summe der Einträge p_tlen, p_dlen, p_blen der BASEPAGE und deren Länge (256). Die Adresse der BASEPAGE wird freundlicherweise von GEMDOS vor dem Start des Programms auf den Stack geschrieben, so daß wir sie mit "move.l 4(sp), dest" erfahren können. Die folgende Vorgehensweise sehen Sie in INSTALL des Beispielprogramms, was in Listing 1 zu finden ist. Die eigentliche residente Speicherung geschieht später.

## Zwei sind einer zuviel

Im weiteren Verlauf des Programms folgt die Initialisierung. Da häufig Initialisierungen auf Speicherbereiche zugreifen, die privilegiert sind - in unserem Beispiel wird auf die Ausnahmevektoren zugegriffen -, wird diese indirekt über die XBIOS-Routine SUPEREXEC aufgerufen. Schauen wir uns (besonders für diejenigen unter Ihnen, die unsere GOODBLIT-Routine letzten Monat schon bewunderten) kurz die Installierung an. Zunächst werden alle Interrupts ausgeschaltet und dann der Originalzeiger des Busvektors geholt. Im nächsten Schritt kommt eine interessante Sache: Es wird geprüft, ob dieser Vektor schon auf unsere (vielleicht schon einmal vorher gestartete) Routine zeigt. Dies geschieht folgendermaßen: Man schreibt vor das Programm irgendeinen Code, in unserem Fall GB10 (Goodblit 1.0). Liest man nun den Startvektor aus, um unser Programm einzuklinken, prüft man, ob dieser Vektor nicht schon auf ein Programm zeigt, das vor der Startadresse diesen Code hat. Ist dies der Fall, so wurde zu einem früheren Zeitpunkt unsere Routine fest im Speicher installiert, und wir können uns jede weitere Installierung sparen, denn was sollen wir mit zweimal demselben Programm im Speicher? Ist das Programm noch nicht im Speicher verankert gewesen, so schreiben wir die Startadresse unseres Programms in den Ausnahmevektor, andernfalls merken wir uns die Reinitialisierung in INSTALLED. War das Programm schon installiert, wird es heimlich, still und leise über die GEMDOS-Routine P_TERM0 verlassen, die den gesamten Speicher des Programmes freigibt. War das Programm noch nicht im Rechner vorhanden, so müssen wir dafür sorgen, daß beim Verlassen des Programms ein gewisser Teil im Speicher nicht freigegeben wird.

## Unsere Routine wird seßhaft

Bei der Programmierung muß man darauf achten, daß die Routinen, die resident im Speicher bleiben sollen, an den Anfang des Programmes und aufeinanderfolgend gelegt werden. Außerdem sollte man beachten, daß kein DATA- und BSS-Bereich verwendet wird, denn dann kann man alle Daten hinter diesen Routinen freigeben. Die Länge wird über die Differenz von Anfang (RESID_A) und Ende (RESID_B) berechnet und das Programm nicht über P_TERM0 sondern über PTERMRES (Gemdos-Routine 49) verlassen. GEMDOS gibt nun allen Speicherplatz ab der Adresse, die sich aus BASEPAGE+Länge_der_residenten_Routinen ergibt, frei. Da sich DATA- und BSS-Segment in dem freigegebenen Bereich befinden, wäre es sehr ungesund (eine bombige Idee, es einmal zu versuchen), auf diesen Bereich zuzugreifen. Wenn Sie unbedingt diese Bereich verwenden wollen, müssen Sie PTERMRES die oben für MSHRINK berechnete Speicherlänge übergeben. Denken Sie dabei aber daran, daß dann ihre von nun an unnützen Initialisierungsroutinen auch resident bleiben. Bevor ich das Thema residente Programme beende, möchte ich sie noch darauf hinweisen, daß die Routinen durch das oben beschriebene Verfahren natürlich nicht RESETFEST sind. Dazu gehört schon etwas mehr Aufwand, "aber das ist eine andere Geschichte, und davon möchte ich Euch das nächste Mal erzählen....".
Zum Schluß unserer heutigen ST-Ecke habe ich noch eine erfreuliche Meldung zu verkünden: Ein Leser hat sich die Mühe gemacht, die in der Mai-Ausgabe der ST-Computer veröffentlichte WORK_TREE-Routine in PASCAL umzuschreiben. Diese Routine, die von Herrn Reichel aus Karlsruhe geschrieben wurde, möchte ich den PASCAL-Freunden unter unseren Lesern nicht vorenthalten. Für weitere Informationen bitte ich Sie, in der Mai-Ausgabe nachzuschlagen. Nur eine kleine Randbemerkung für alle diejenigen unter Ihnen, die diese ST-Ecke gelesen haben: Beachten Sie, daß in der allgemeinen Darstellung der WORK_TREE-Routine eine SOLANGE-WIE-Schleife (in C bedeutet dies DO-WHILE) verwendet wurde, die es in PASCAL nur in der Form SOLANGE-BIS (REPEAT-UNTIL) gibt. Dabei muß, um Gleichheit zu erreichen, die Schleifenbedingung umgedreht werden. Das bedeutet, daß aus "Solange,wie ob_next ungleich -1" ein "Solange, bis ob_next gleich -1 " wird. Dieser kleine Fehler ist Herrn Reichel unterlaufen. Trotzdem bedanke ich mich vielmals bei ihm für die Mühe, die er sich gemacht hat. Die PASCAL-Routine finden Sie in Listing 2. Das Beispiel ist so ausgelegt, daß die Resource-Datei geladen wird. Viel Spaß und Erfolg beim Einsatz dieser Routine, und wie wäre es zum Beispiel mit einer WORK_TREE-Routine in MODULA-2?

*Stefan Höhn*

*Listings folgen.*

*Listing 1: Goodblit-Programminitialisierung*

```
  1: *
  2: *        --------------------
  3: *        |    GOODBLIT      |
  4: *        --------------------
  5: *
  6: *        BLITTER Fehler-Korrektur
  7: *
  8: *        ASM source code der Initialisierung
  9: *        der gesamten Source-Code ist in ST-Computer
 10: *        Ausgabe Juni 1988 zu finden.
 11: * ---------------------------------------------------
 12: *
 13: *        (c) 1988 by IMAGIC Grafik GbR.
 14: *
 15: *        geschrieben  13-2-88 by Jörg Drücker.
 16: *
 17: *===================================================
 18: *
 19: *     Dieses Programm installiert sich selbst RESIDENT
          im Speicher
 20: *     ----------------------------------------------
 21: *
 22: *     Anmerkung: Dieses Programm ist für den AUTO-Ordner
 23: *     gedacht, kann aber auch direkt von DESKTOP aus
          gestartet werden.
 24: *
 25: *===================================================
 26: *
 27: *     OS-Konstanten
 28: *
 29: *     TRAPS:
 30: GEMDOS:    equ        1
 31: BIOS:      equ        13
 32: XBIOS:     equ        14
 33: *
 34: *     GEMDOS:
 35: P_TERM0:   equ        0
 36: C_CONWS:   equ        9
 37: P_TERMRES: equ        $31
 38: M_SHRINK:  equ        $4A
 39: *
 40: *     XBIOS:
 41: SUPEREXEC: equ        38
 42: *
 43: *-----------------------------------------------*
 44: * Der Bereich von RESID_A bis RESID_B bleibt resident
 45: *-----------------------------------------------*
 46:
 47: RESID_A:    bra       INSTALL
 48:
 49: *--------------------------------------*
 50: *
 51: *      Hauptprogramm
 52: *      -------------
 53: *
 54: *--------------------------------------*
 55:
 56:        dc.b       'GB10'  * "installiert" Merker
 57:
 58: MAIN:
 59:
 60: * .... hier kommt das Hauptprogramm hin....
 61:
 62: RESID_B:
 63: *----------------------------------*
 64: *                                  *
 65: * Installierung                    *
 66: *                                  *
 67: *----------------------------------*
 68:
 69: INSTALL: move.l   sp,a6      * System-Stack-Pointer
 70:          lea      USTACK,sp  * Installiere Benutzer-
                                   Stack
 71:
 72:          move.l   4(a6),a6   * Adresse der BASEPAGE
 73:          move.l   $C(a6),a4  * TEXT Segment-Länge
 74:          adda.l   $14(a6),a4 * + DATA Segment-Länge
 75:          adda.l   $1C(a6),a4 * + BSS Segment-Länge
 76:
 77:          pea      256(a4)    * + BASEPAGE-Länge
 78:          pea      (a6)         * BASEPAGE
 79:          clr.w    -(sp)        * dummy
 80:          move.w   #M_SHRINK,-(sp) * verkleinere
                                     benutzten Speicher
 81:          trap     #GEMDOS
 82:          lea      12(sp),sp  * Stack korrigieren
 83:
 84:          pea      INSTAL_GOODBLIT(pc) * installiere
                                     GOODBLIT
 85:          move.w   #SUPEREXEC,-(sp)
 86:          trap     #XBIOS           * In Supervisor-
                                     Mode ausführen
 87:          addq.l   #6,sp
 88:
 89:          move.w   INSTAL_FLAG(pc),d0  * Klappte
                                     Installierung
 90:          bne.s    ITS_DONE        * JA, Text ausgeben,
                                   resident
 91:                                   * bleiben
 92:          move.w   #P_TERM0,-(sp) * Programm ohne
                                  Anmerkung verlassen
 93:          trap     #GEMDOS
 94:
 95: *=====================================*
 96:
 97: ITS_DONE: pea     ID(pc)
 98:          move.w   #C_CONWS,-(sp)
 99:          trap     #GEMDOS
100:          addq.l   #6,sp
101:
102:          clr.w    -(sp)             * Returnwert 0
103:          pea      RESID_B-RESID_A+256 * minimale
                                    Resident-Länge
104:          move.w   #P_TERMRES,-(a7)
105:          trap     #GEMDOS
106:
107: *------------------------------------*
108:
109: INSTAL_GOODBLIT:
110:          ori.w    #$700,sr      * IPL 7, Interrupts
                                    ausschalten
111:
112:          lea      8,a1          * Bus-Fehler-Zeiger
113:          movea.l  (a1),a0       * Original-Wert holen
114:
115:          cmpi.l   #'GB10',-(a0) * GOODBLIT schon
                                    installiert ?
116:          beq.s    INSTALLED     * scheint so !
                                    -> Programmende
117:
118:          lea      JMP_BUSERR+2(pc),a0 * Adresse der
                                    Modifikation
119:          move.l   (a1),(a0)     * Alten Vektor in
                                    Befehl abspeichern
120:          move.l   #MAIN,(a1)    * und uns in den
                                    Zeiger schreiben
121:
122:          st       INSTAL_FLAG   * "Installiert"
                 -Merker setzen
123:
124: INSTALLED:
125:          rts
126:
127: *----*
128: INSTAL_FLAG: dc.w     0            * Enthält FF, wenn
                                    schon installiert
129:
130: *------------------------------------*
131: * Ausgabe-Text
132:
133: ID:      dc.b     $1B,'E'
134:          dc.b     'The Unknown ',$BD,' 1988 by Me'
135: END_ID:  dc.w     0
136:
137: *------------------------------------*
138:
139:          bss
140:
141:          ds.b     $100
142: USTACK:  ds.w     0
143:
144:          end
```

*Listing 2: Worktree-Routine in PASCAL*

```
 1: {$P-}                  (* UNBEDINGT NOTWENDIG *)
 2: { 'schützt' vor Überlauf des Stacks bei der Rekursion }
 3:
 4: program worktest;
 5:
 6: const
 7: {$I GEMCONST.PAS }
 8:     Wurzel =0;
 9:
10: TYPE
11: {$I GEMTYPE.PAS }
12:
13: VAR
14:     bau  : Dialog_Ptr;
15:     s    : String[10];
16:     p    : Tree_index;
17:     junk: Integer;
18:
19: {$I GEMSUBS.PAS }
20:
21: FUNCTION Select_Objc( box : Dialog_Ptr;
                         item : integer ) : integer;
22: BEGIN
23:   Obj_SetState(box,item,SELECTED,FALSE);
24: END;
25:
```

## Public-Domain Software

Wir liefern auf erstklassigem Diskettenmaterial die PD-Software **1- heute** dieses Atari ST Magazins sowie eigene, nur bei uns erhältliche PD-Programme!

jede Diskette nur **DM 8,--**

**PD - 10er-Blöcke**

10 PD-Programme auf jeweils 5 Disketten erhalten Sie von unserem Kopierservice für nur

(je Block) **DM 40,--**

**Neu!**
**MS-DOS Freesoftware**

für Besitzer eines Atari ST mit MS-DOS-Emulator (PC-Ditto)!

jede Diskette nur **DM 8,--**

Fordern Sie unsere **PD-Listen** an! Bitte schriftlich mit Angabe Ihres Computertyps!

## Technobox-Produkte

Campus CAD V1.3

Maschinen- und Werkzeugbau, Fertigungs- und Elektrotechnik sowie Architektur, einfach alles, was mit technischen Zeichnen zu tun hat, meistert diese neue Campus-Version professionell!

| | |
|---|---|
| | **nur 798,--** |
| Demoversion | 40,-- |
| **Neu! Zeichnungsprogramme** | |
| Campus ART | 149,-- |
| Campus Draft | 149,-- |

**Einzelinfo anfordern (schriftlich!)**

## Diskettenlaufwerke

• Erstklassige Verarbeitung • komplett mit Metallgehäuse, Netzteil (eingebaut oder Stecknetzteil) und Kabel • leise • anschlußfertig und vollkompatibel •

| | | |
|---|---|---|
| 3,5" Floppy | (720 KB) | **348,--** |
| 3,5" Floppy | (Stecknetzteil) | **298,--** |
| 3,5" Doppelfl. | (1,4 MB) | **648,--** |
| 3,5" Doppelfl. | (Stecknetzteil) | **598,--** |
| 5,25 Floppy | (720 KB) | **448,--** |
| 3,5"+5,25" Fl. | (1,4 MB) | **798,--** |

## Handy-Scanner

**mit Grafikpaket CAMERON HANDY PAINTER 2.0 und deutschem Handbuch.** (bei Atari nur s/w M.)

- Einsatz im Desktop-Publishing
- Abrastern von Bildern (f oder sw)
- Speichern von Unterschriften
- Anfertigen von illustrierten Handbüchern
- Überarbeiten und Entwickeln von Logos und Typen

| | |
|---|---|
| Schwarz-Weiß Version | **798,--** |
| Typ 3 mit Graustufen | **848,--** |

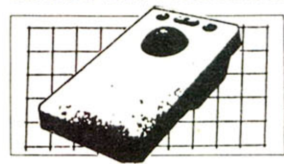
**Einzelinfo anfordern (schriftlich!)**

## Atari-Schaltpläne

| | | |
|---|---|---|
| 260 ST / 520 ST | | **29.80** |
| 520 ST+ / 520 STM | | **29.80** |
| 1040 STF | | **29.80** |
| 1040 Erweiterung | | **29.80** |
| SF 314 / SF 354 | je | **19.80** |
| SNM 804 / 1050 | je | **19.80** |
| 600 XL / 800 XL | je | **19.80** |
| SC 1224/SM 124 | je | **19.80** |

## Marconi RB2 Trackerball

Die Maus ist tot, es lebe der Trackerball



Der Marconi Trackerball eignet sich hervorragend im CAD/CAM Bereich, in der Textverarbeitung und zur Positionierung des Cursors auf dem Bildschirm. Dank seines kompakten Gehäuses benötigen Sie keinen Platz mehr zum Bewegen einer Maus.

**Händleranfragen erwünscht!**
**Einzelinfo schriftlich anfordern!**
(Angabe des Computertyps)

**nur 198,--**

# Alles aus einer Hand!

# Software

### APPLICATION SYSTEMS:

| | |
|---|---|
| Signum 2 | 448,-- |
| Signum-Fontdisketten: | |
| Fontdiskette Julia | 100,-- |
| Eurofont Diskette | 69,-- |
| Professional Fontd. | 100,-- |
| Fontdisk. Rokwel | 100,-- |
| Signum! Utility | 89,-- |
| STAD | 178,-- |
| Megamax C-Comp. | 398,-- |
| deutsches Handb. | 49,-- |
| Megam. Modula 2 | 398,-- |
| Imagic | 498,-- |
| FlexDisk (Ramdisk) | 69,-- |
| Harddisk Utility | 69,-- |
| Editor Toolbox | 149,-- |
| Bolo (Superspiel) | 69,-- |

### Kieckbusch:

| | |
|---|---|
| Timeworks DTP | 389,-- |
| VIP Professional | 299,-- |
| STEVE | 398,-- |
| STEVE-Lehrbuch | 58,-- |
| LOGiSTiX | 398,-- |

| | |
|---|---|
| A-MAGIC Turbo Dizer | 298,-- |
| Multi-Hardcopy | 98,-- |
| Desk Assist II+ | 139,-- |

### GFA Produkte:

| | |
|---|---|
| GFA Farbkonverter | 59,-- |
| GFA Monochromkonvert. | 59,-- |
| GFA GEM-Autostarter | 59,-- |
| GFA Floppy-Speeder | 59,-- |
| GFA Vektor | 99,-- |
| GFA Starter | 59,-- |
| monoSTar PLUS | 149,-- |
| ST DIGI-DRUM | 79,-- |
| GFA BASIC Comp. | 99,-- |
| GFA BASIC Interpr. | 99,-- |
| GFA-Assembler | 149,-- |
| GFA-Raytrace | 149,-- |
| GFA Objekt | 198,-- |
| GFA DRAFT | 198,-- |
| GFA DRAFT plus | 349,-- |
| GFA MOVIE | 149,-- |
| GFA ARTIST | 149,-- |
| GFA Publisher | 398,-- |
| GFA BAS. 68881 | 349,-- |
| **Bücher:** GFA Basic | 79,-- |
| GFA TOS& GEM | 49,-- |
| GFA BASIC Progr. | 49,-- |

### Aladin 398,--

**Mac-Betriebssystem**
(ROM-Satz) **195,–**

### Heim Produkte:

| | |
|---|---|
| Salix Prolog | 198,-- |
| Steuer Tax V2.7 | 98,-- |
| ST Aktie | 69,-- |
| ST Plot/ST Print je | 59,-- |

### Tommy Software:

| | |
|---|---|
| 1ST Speeder | 89,-- |
| MusiX32 | 89,-- |
| 1ST Freezer | 148,-- |
| Dizzy Wizard | 69,-- |

### KUMA:

| | |
|---|---|
| K-Spell | 118,-- |
| K-Seka | 168,-- |
| K-Spread 2 | 198,-- |
| K-Graph 2 | 148,-- |
| K-Com | 148,-- |
| K-Resource | 118,-- |
| K-Word 2 | 118,-- |
| K-RAM | 89,-- |
| K-Switch | 99,-- |
| K-Minstrel | 89,-- |

### G-Data Produkte:

| | |
|---|---|
| G-Ramdisk II | 48,-- |
| G-Diskmon II | 98,-- |
| Harddisk Help & Extension | 129,-- |
| Interprint II | 49,-- |
| Interprint II Ramdisk | 99,-- |
| AS Sound - Sampler II (Softw.) | 198,-- |
| Sampler III (16 BIT) | 598,-- |
| Relas | 398,-- |
| Sympatic Paint | 298,-- |
| TV Modulatoren: | |
| mit Monitorumschalter | 249,-- |
| ohne Monitorumschalt. | 239,-- |
| Retrace Recorder | 99,-- |
| Disk Help | 79,-- |
| Fast Speeder | 129,-- |
| Disk Help | 79,-- |
| G-Datei | 199,-- |
| M.A.R.S. | 129,-- |
| Chess | 129,-- |
| G Copy | 99,-- |
| G Scanner | 298,-- |

### Omicron Produkte:

| | |
|---|---|
| Omicr. Basic (Disk.) | 179,-- |
| Omicr. Bas. (Modul) | 229,-- |
| Omicr. Compiler | 179,-- |
| Omicr. Assembler | 99,-- |

### Desktop Publishing

| | |
|---|---|
| Calamus | 998,-- |
| Publ. Partner | 498,-- |
| Fleet Str. Edit. | 348,-- |
| Timeworks Publisher | 389,-- |

### Royal Prod.:

| | |
|---|---|
| Disk-Royal | 89,-- |
| Katpro-Royal | 89,-- |
| Wizard-Royal | 139,-- |
| Voc-Royal | 79,-- |

### Text:

| | |
|---|---|
| 1st Word | 99,-- |
| 1st Word plus | 199,-- |
| 1st Proport. | 95,-- |
| Wordstar | 199,-- |
| Textomat ST | 99,-- |
| Protext ST | 148,-- |
| Becker Text | 199,-- |
| Megafont ST | 119,-- |
| Word Perfect | 910,-- |
| Typeset. Elite | 139,-- |

### Grafik:

| | |
|---|---|
| Art Direktor | 173,-- |
| Film Direktor | 188,-- |
| Degas | 169,-- |
| MICA | 298,-- |
| Animator | 119,-- |
| Campus | 798,-- |

### MCC:

| | |
|---|---|
| Lattice C omp. | 298,-- |
| Pascal Comp. | 248,-- |
| Makro Assembl. | 169,-- |
| MCC Make | 169,-- |
| MCC Lisp | 448,-- |
| BCPL-Comp. | 329,-- |
| Pro Fortran | 448,-- |
| Pro Pascal | 448,-- |
| Modula II | 448,-- |
| Copystar | 169,- |

### Datenbanken

| | |
|---|---|
| Adimens ST | 199,-- |
| Aditalk | 189,-- |
| DB Man dtsch. | 395,-- |
| Logistix | 398,-- |
| Profimat ST | 99,-- |

## HEIM Produkte

| | |
|---|---|
| **Bücher:** Omicron Basic | 29,-- |
| Progr. in Omicron Basic | 49,-- |
| Das große VIP-Buch | 49,-- |
| C auf dem Atari ST | 49,-- |
| Anwend. in GFA-Basic | 49,-- |
| GFA-Basic Prg.samml. | 49,-- |
| **GFA-Basic 3.0 Buch** | **59,--** |
| **1st Word Plus Buch** | **49,--** |

| | |
|---|---|
| **Software:** ST Archivar | 89,-- |
| ST Print (4 nützliche Progr.) | 59,-- |
| ST Plot (Kurvendiskussion) | 59,-- |
| ST Aktie (Aktienverwaltung) | 69,-- |
| ST Digital (Logiksimulator) | 89,-- |
| Skyplot plus (Astronomiepr.) | 198,-- |
| ST-Learn (Vokabeltrainer) | 59,-- |
| ST Strukturpainter Strukturprg. zum Erstellen von Graphiken | 89,-- |
| TKC-Faktura ST Integriertes Softwarepaket für den Handel | 899,– |
| TKC-Einnahme /Überschuß ST Buchführung für Freiberufler | 149,– |
| TKC-Haushalt ST Haushaltsbuchführung | 129,– |

## Zubehör

| | |
|---|---|
| **Weide - Produkte:** | |
| Echtzeituhr | 129,– |
| Speichererweiterung | a. Anfr. |
| Video Sound Box Anschluß Ihres ST's an Farbfernseher | 298,– |
| **Abdeckhauben:** | |
| Konsole 520/1040 | 29,-- |
| Monitor (124/1224) | 39,-- |
| CSF-Gehäuse für 260/520 | 148,– |
| **ROM-Satz** für alle ST's | 168,– |
| Akustikkoppler 300 | 278,– |
| Akustikkoppler 300/1200 | 378,– |
| Konzepthalter | 24,80 |
| **Trackball** org. Atari | 98,-- |
| **Monitor-Box** umschaltbar Monochrom / Color | 58,-- |
| **Mouse Pad** Gleitmatte für Maus | 19,80 |
| **Etiketten** endl., 70x70 (200 Stk) | 16,-- |
| **Pal Interface III** | 198,– |

Literatur von **Markt & Technik** und **Data Becker**. Liste anfordern!

## PC DITTO 178,--

MS-DOS Softwareemulation für Ihren Atari-ST • Für Farb- und Monochrom-Monitore • Stützt Festplatten sowie seriell und parallel angeschlossene Drucker • Mit leichtverständlichem deutschem Manual.

**TIM 1.1 298,--**

Time is Money, die Buchführung für den Atari ST (Test Atari ST 7/87).

**TEMPUS Editor**

Tempus hilft dem Hobby- und dem professionellen Programmierer, kostbare Zeit einzusparen! **109,--**

## HEIM MANAGER

Der Heim Manager ist ein einfach zu bedienendes Programm für den privaten Haushalt. Es erlaubt die Überwachung Ihrer Finanzsituation, Textverarbeitung, Adressverwaltung und hilft Ihnen bei verschiedenen Rechenproblemen! Mit 50-seitigem Handbuch.

**Inhalt:** Adressverwaltung • Haushaltsbuch • Terminplaner • Textverarbeitung • Rechner •

Das Programm läuft unter GEM, Atari ST, SW-Monitor, Maus, Rom-TOS.

komplett für **nur 98,--**

Karl-Heinz Weeske • Potsdamer Ring 10 • 7150 Backnang • Telex 724410 weeba d • Kreissparkasse Backnang (BLZ 60250020) 74397 • Postgiro Stgt. 83326-707 • FAX 60077

7-88

**WEESKE**
COMPUTER-ELEKTRONIK

Zahlung per Nachnahme oder Vorauskasse (Ausland per Scheck). Versandkostenpauschale (Inland 6,80 DM / Ausland 16,80 DM). Bitte bei Bestellung + Infoanforderung den Computertyp angeben!

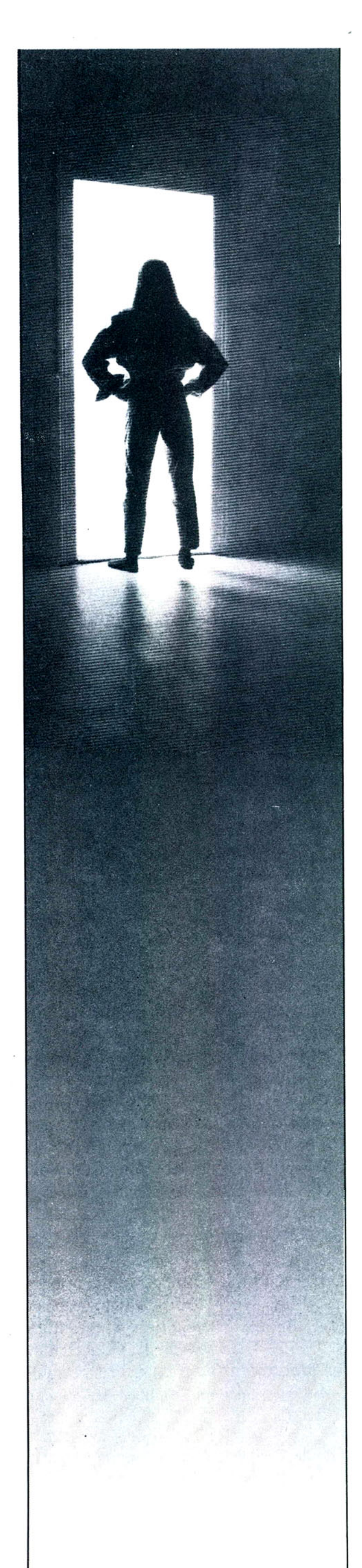
**07191/1528-29 od. 60076**

```
26:  FUNCTION Deselect_Objc( box : Dialog_Ptr;
                            item : integer) : integer;
27:  BEGIN
28:    Obj_SetState(Box,item,NORMAL,FALSE);
29:  END;
30:
31:  FUNCTION Pr_baum( box : Dialog_Ptr;
                      item : integer) : integer;
32:  BEGIN
33:    writeln('item: ',item);
34:  END;
35:
36:  {--------------------------------------------------}
37:  {                                                  }
38:  {                 W O R K _ T R E E                }
39:  {                 = = = = = = = = =                }
40:  {                                                  }
41:  { Übertragung der C-Implementierung z. Durchwanderung
       der Objekt-    }
42:  { struktur eines Baumes in PASCAL 2.0 - Aus ST-Computer
       5/88 S. 145    }
43:  { Volker REICHEL 7500 Karlsruhe 41 Durlacherstr. 23
       4. Mai 1988    }
44:  {                                                  }
45:  {**************************************************}
46:  { work_tree wandert die Objektstruktur eines Baumes ab
       und führt dabei  }
47:  { eine Routine aus, die das aktuelle Objekt bearbeitet.
       Dieser Routine   }
48:  { werden die Baumadresse und der Index des momentan
       aktuellen Objekts   }
49:  { übergeben. 'Achtung': work_tree ruft sich rekursiv
       auf...          }
50:  {                                                  }
51:  { PARAMETER: baum     - Baumadresse                }
52:  {            ob_an    - Anfang der Wanderung       }
53:  {            ob_en    - Ende der Wanderung         }
54:  {            aufruf   - Routine, die ausgeführt werden
                             soll                       }
55:  {                                                  }
56:  { Soll der gesamte Baum bearbeitet werden, werden ob_an
       und ob_en auf    }
57:  { Null gesetzt.                                    }
58:  {**************************************************}
59:
60:  PROCEDURE work_tree( Var baum : Tree_Ptr; ob_an,ob_en :
                          Tree_Index;
61:           function aufruf( t : Tree_Ptr; ob :
                                 Tree_Index): integer );
62:  LABEL 99;
63:  VAR
64:         ob_ind : Tree_Index;
65:         junk   : Integer;
66:  BEGIN
67:
68:    ob_ind := ob_an; { aktueller Objektindex = 1. Objekt }
69:
70:    REPEAT              { ENDLOS bis EXIT ! }
71:
72:      junk := aufruf(baum,ob_ind); { Aufruf der Routine }
73:        IF baum^[ob_ind].ob_head <> Null_Index THEN { falls
                                                 Verzweigung
74:                                              vorhanden }
75:          work_tree(baum,baum^[ob_ind].ob_head,
                   baum^[ob_ind].ob_tail,aufruf);
76:
77:        IF ob_ind = ob_en THEN
78:          GOTO 99;            { letztes Objekt bearbeitet }
79:        ob_ind := baum^[ob_ind].ob_next; { nächstes Objekt }
80:
81:      UNTIL baum^[ob_ind].ob_next = Null_Index; { bis Ende
                                          des Unterobjektes }
82:
83:  99:
84:  END;   { work_tree }
85:
86:  {--------------------}
87:  {   Hauptprogramm    }
88:  {--------------------}
89:
90:  BEGIN
91:    if init_gem >= 0 then
92:    BEGIN
93:     IF load_resource('BAU.RSC') THEN
94:     BEGIN
95:      Find_Dialog(Wurzel,bau);
96:      center_Dialog(bau);
97:
98:      junk := Do_Alert('[1][Dialog-Baum ist eingelesen]
                                                 [OK]',1);
99:      p := do_dialog(bau,0);
100:
101:     work_tree(bau,0,0,Select_objc);
102:     p := do_dialog(bau,0);
103:     junk := Do_Alert('[1][Alle Objekte wurden
                         selektiert][OK]',1);
104:
105:     work_tree(bau,0,0,Deselect_objc);
106:     p := do_dialog(bau,0);
107:     junk := Do_Alert('[1][Alles wieder deselektiert]
                                                 [OK]',1);
108:
109:     work_tree(bau,0,0,Pr_baum);
110:     junk := Do_Alert('[3][Das war''s!][OK]',1);
111:
112:     End_Dialog(bau);
113:     Delete_Dialog(bau);
114:     exit_gem;
115:    END;
116:   END;
117:  END.
```

*Listing 3: Fehlende Zeilen der Goodblit-Routine aus ST 6/88*

```
258: move.l    (a1),(a0)        * Alten Vektor in Befehl abspeicher
259: move.l    #GOODBLIT,(a1)   * und uns in den Zeiger schreiben
260:
261: st        INSTAL_FLAG      * "Installiert"-Merker setzen
```

***ENDE***



# AUF DER SCHWELLE ZUM LICHT

## Directory-Verwaltung Teil 1

**Heute geht es um die Directory-Verwaltung. Da dies wieder ein sehr umfangreiches Kapitel des GEMDOS ist, haben wir es in zwei Teile aufgeteilt. Im ersten Teil beschäftigen wir uns mit der internen Verwaltung von Directories (Dateiverzeichnissen), Pfaden und den zugehörigen GEMDOS-Funktionen. Im zweiten Teil, der in der nächsten Ausgabe erscheint, geht es dann um Suchoperationen in Directories. Diesmal kommen die Freunde der allseits beliebten GEMDOS-Fehler wieder voll auf ihre Kosten.**

## Allgemeines

Zunächst ein paar allgemeine Bemerkungen zur Struktur der Directories (Dateiverzeichnisse) unter GEMDOS. Jedes Laufwerk verfügt über ein Root Directory (Wurzelverzeichnis), welches mit '\' angesprochen wird. Das Root Directory kann neben "normalen" Anwenderdateien wie Programmen, Texten usw. weitere (Sub-)Directories (Ordner) enthalten, die eigene Namen haben. Jeder Ordner kann selbst wiederum weitere Ordner enthalten, so daß sich eine hierarchische Struktur ergibt.

Ein übergeordnetes Directory wird "Parent" genannt, ein Subdirectory ist ein "Child". Diese Bezeichnungen sind jeweils relativ zu einem anderen Directory gemeint, d.h., ein Child kann selbst wiederum Parent eines anderen Directories sein.

Dateien werden durch ihren Zugriffspfad bezeichnet, der sich aus dem Laufwerk und der Abfolge von durchlaufenen Ordnern vom Root Directory bis zum Directory, in dem sich die Datei befindet, ergibt (z.B. 'A:\ MEGAMAX\ HEADERS\ STDIO.H'). Dies ist jedem Anwender vom Arbeiten mit dem Desktop her bekannt.

Bei den dem Programmierer zur Verfügung stehenden GEMDOS-Funktionen können außer diesen vollständigen Dateipfaden auch nur bestimmte Teile davon angegeben werden. Beim Fehlen der Laufwerkskennung bezieht sich der Pfad auf das sogenannte "Standardlaufwerk", welches mit 'Dsetdrv' festgelegt werden kann (z.B. '\ MEGAMAX\ HEADERS\ STDIO.H').

Wenn der erste Backslash ('\') fehlt, bezieht sich die Ordnerfolge nicht auf das Root Directory, sondern wird an den "Standardpfad", der mit 'Dsetpath' eingestellt wird, angehängt (relative Pfadangabe). Z.B. meint bei einem Standardpfad '\MEGAMAX\' für Laufwerk A: der Name 'A:HEADERS\STDIO.H' die vollständige Bezeichnung 'A:\ MEGAMAX\ HEADERS\ STDIO.H'. Diese beiden Abkürzungsmöglichkeiten können auch miteinander kombiniert werden, z.B. könnte beim letzten Beispiel auch nur 'HEADERS\ STDIO.H' angegeben werden, wenn A: Standardlaufwerk ist.

Diese letzte Möglichkeit ist in den meisten Fällen zu bevorzugen, da hierbei die absolute Position einer Datei im Dateisystem nicht bekannt zu sein braucht.

Bei der kompletten Angabe von Dateinamen, wie in vielen Programmen leider der Fall, besteht die Gefahr, daß eine Datei z.B. immer nur auf Laufwerk A: gefunden wird, obwohl es sich auf C: oder in einem anderen Ordner befindet.

Weiterhin können die speziellen Directories '.' und '..' verwendet werden. '.' bezeichnet das aktuelle Directory selbst

und '..' führt eine Directory-Ebene höher. So ist z.B. 'HEADERS\ .\ STDIO.H' identisch mit 'HEADERS\ STDIO.H' und '..\ ERRORS.OUT' bezeichnet '\MEGAMAX\ ERRORS.OUT', wenn der Standardpfad wieder '\MEGAMAX\ HEADERS\' ist.
GEMDOS verwaltet für jedes Laufwerk einen eigenen Standardpfad. Dabei hat sogar jeder Prozeß seine eigenen Standardpfade und Standardlaufwerke. Sie werden an nachgeladene Child-Prozesse "weitervererbt". Die Pfade des Parent-Prozesses können vom Child-Prozeß nicht verändert werden.
Bei der maximalen Schachtelungstiefe von Ordnern gibt es keine prinzipielle Begrenzung von Seiten des GEMDOS aus. In der Praxis gibt es Einschränkungen bei der maximalen Pfadlänge (s. 'Dgetpath') und bei der "internen Speicherverwaltung".

## Aufbau der Directories

Subdirectories sind ganz ähnlich wie "normale" Dateien aufgebaut. Ihr Directory-Eintrag in ihrem Parent Directory unterscheidet sich hauptsächlich im Dateiattribut. Bei Directories ist hier Bit 4 gesetzt. Außerdem ist als "Dateilänge" immer Null eingetragen. Die Directory-Datei besteht aus lauter 32 Byte großen Directory-Einträgen. Ihre Struktur ist in Abb. 1 dargestellt.

```
typedef struct
{  char dir_name[11];            /* Dateiname          */
   char dir_attr;                /* Attribut           */
   char dir_dummy[10];           /* unbenutzt          */
   unsigned int dir_time;        /* Erstellungszeit    */
   unsigned int dir_date;        /* Erstellungsdatum   */
   int dir_stcl;                 /* erster Cluster     */
   long dir_flen;                /* Dateilänge         */
} DIR;
```

*Abb. 1: Directory-Eintrag*

'dir_name' ist der Dateiname im GEMDOS-Format. Er ist immer genau 11 Zeichen lang, 8 Zeichen für den Hauptteil des Namens, 3 Zeichen für die Namenserweiterung (Extender). Zu kurze Namensteile werden mit SPACE aufgefüllt ('DATEI.X' wird hier als 'DATEI X ' dargestellt). Jeder Dateiname darf pro Directory nur einmal vorkommen. Es darf also keine Anwenderdatei und einen Ordner gleichen Namens geben. 'dir_attr' ist das "Dateiattribut", das Auskunft über Typ und einige Eigenschaften der Datei gibt. Es wird nächsten Monat noch diskutiert.
Der timestamp (Zeit und Datum der Erstellung der Datei) findet sich in 'dir_date' und 'dir_time' (zur Repräsentation dieser Daten siehe spätere Folge über die "Timer-Funktionen").
Die Nummer des ersten Clusters der Datei ist 'dir_stcl'. Die weiteren Clusternummern werden mit der FAT ermittelt (s. März-Ausgabe). 'dir_flen' schließlich ist die Länge der Datei in Bytes.
Timestamp, Startcluster und Dateilänge sind übrigens im Intel-Format gespeichert, d.h. die Reihenfolge von höher- und niederwertigen Bytes ist gegenüber dem Motorola-Format gerade vertauscht.
Beim Löschen von Dateien wird im Directory nur das erste Zeichen des Dateinamens auf $E5 gesetzt, der Rest bleibt erhalten. Da die zugehörigen Cluster in der FAT aber freigegeben werden, ist ein Rekonstruieren gelöschter Dateien nicht so ohne weiteres möglich.
Ein neu angelegtes Directory besteht zuerst aus einem Cluster, der anfangs gelöscht ist (nur Nullbytes enthält). Sobald mehr Dateien erzeugt werden, als hier Platz haben, wird das Directory automatisch um einen Cluster verlängert, der ebenfalls zuerst gelöscht wird.
Eine Ausnahme bildet wiederum das Root Directory. Es hat eine fest vorgegebene Länge, die nicht überschritten werden kann. Das Root Directory wird beim Formatieren komplett mit Nullen gefüllt.
Ein Directory wird jedoch nie verkürzt, auch dann nicht, wenn der letzte Cluster überhaupt nicht mehr benutzt wird.
Neue Einträge werden immer an der ersten freien Position gemacht. Die Einträge haben somit keine bestimmte Reihenfolge. Für die sortierte Anzeige in den Fenstern ist der Desktop verantwortlich.
Jedes Subdirectory hat noch zwei spezielle Einträge, die die Dateinamen '.' und '..' haben. Sie werden bei Anlegen eines Subdirectory automatisch erzeugt und erscheinen im allgemeinen bei der Anzeige von Directories nicht. '.' verweist auf das Directory selbst ('dir_stcl' ist der eigene Start-Cluster), '..' führt zum Parent Directory ('dir_stcl' ist der Start-Cluster des Parent Directories). Wenn das Root Directory der Parent ist, enthält 'dir_stcl' hier Null.

## Directory-Dateien

GEMDOS verwaltet Directories ganz ähnlich wie Anwenderdateien. Das bedeutet, jedes Directory hat einen eigenen "File Descriptor" (FD), über den die eigentlichen Datei-Zugriffe wie Lesen, Schreiben und Positionieren laufen. Dabei finden die allgemeinen internen Routinen Verwendung, wie sie in der März-Folge beschrieben wurden. Für die von GEMDOS intern verwalteten Directory-Dateien gibt es keine Datei-Handles und keine "File Control Blocks" (FCB).
Da auf Directories wesentlich häufiger zugegriffen werden muß als auf andere Dateien, wird hier ein erhöhter Verwaltungsaufwand betrieben. Dies hat zur Folge, daß '.' und '..' von GEMDOS überhaupt nicht benutzt werden. Sie existieren nur aus Kompatibilitätsgründen zu PC-DOS.
GEMDOS merkt sich alle Verzeichnisse, auf die irgendwann einmal zugegriffen wurde, um bei weiteren Suchoperationen sofort zu wissen, was mit dem Directory los ist. Dies spart erheblich an Zugriffen auf die übergeordneten Directories. Zu jedem GEMDOS bekannten Directory existiert dazu eine Datenstruktur, die ich "Directory Descriptor" (DD) getauft habe. Die DDs sind in einer Baum-Struktur entsprechend der Directory-Hierarchie angeordnet. Jedesmal, wenn eine der GEMDOS-Dateifunktionen ein Directory anspricht, für das es noch keinen DD gibt, wird dieser automatisch angelegt und in den Baum einsortiert. GEMDOS geht sogar noch weiter, denn das Bekanntmachen neuer Directories geschieht sozusagen im Vorbeigehen. Bei jeder Suchoperation (also auch beim Suchen gewöhnlicher Dateien) werden nämlich alle Subdirectories, deren Einträge untersucht werden, bekannt gemacht. Da sich GEMDOS nicht für '.' und '..' interessiert, gibt es dafür auch keine DDs.

## "Directory Descriptoren" (DD)

DDs sind nun folgendermaßen aufgebaut (s. Abb. 2). Ähnlich wie die FDs beinhalten sie einige Informationen zum Eintrag des Directories in dessen Parent Directory. Dazu gehören der eigene Name im GEMDOS-Format ('dd_name'), der erste Cluster ('dd_stcl') sowie Erstellungszeit und -datum

('dd_time' und 'dd_date'). Der Startcluster liegt in GEMDOS-Zählung und der timestamp im Intel-Format vor. Für weitere Erläuterungen vergleichen Sie bitte mit den Erklärungen zum FD aus der letzten Folge.
'dd_dfd' zeigt auf den FD des Directories und bedarf wohl keiner weiteren Erklärung. NIL zeigt an, daß das Directory zwar schon erkannt, aber noch kein FD benötigt wurde. Der FD wird erst dann eingerichtet, wenn auch tatsächlich ein Zugriff auf das Directory stattfindet. Auch eine Referenz auf das zugehörige Laufwerk muß sein: 'dd_dmd' verweist auf den "Drive Media Descriptor", der ja auch schon früher behandelt wurde.
Im letzten Artikel wurde erwähnt, daß die FDs aller geöffneten Dateien eines Directories in einer Liste verknüpft sind. 'dd_fdl' zeigt auf den Beginn dieser Liste. Dabei bedeutet NIL, daß keine Dateien geöffnet sind.
Doch kommen wir nun zum eigentlich interessanten Teil, der Verkettung der einzelnen DDs untereinander (Abb. 3). Der Rückverweis auf das eigene Parent Directory geschieht mit 'dd_pardd'. Das Root Directory hat als einziges keinen Parent; daher steht hier NIL. 'dd_chdd' bezeichnet ein Child Directory; wenn noch keins bekannt ist, steht hier NIL. Die einzelnen Directories einer Hierarchie-Ebene sind in einer 'Child-Liste' mit 'dd_link' verbunden, das Ende der Liste wird durch NIL markiert. Die Reihenfolge der Directories in der Liste ist im allgemeinen umgekehrt wie in ihrem Parent. Das ist aber nicht immer so, daher ist diese Reihenfolge als undefiniert anzusehen.
Mit Hilfe dieser drei Zeiger kann sich GEMDOS also relativ schnell innerhalb der Directory Hierarchie bewegen, ohne daß Zugriffe auf den Massenspeicher notwendig sind.
Da Zugriffe auf das Parent Directory ziemlich häufig sind, gibt es noch weitere Verknüpfungen zwischen Parent und Child. In der Abb. 4 sind alle Beziehungen eines Child Directories zu seinem Parent dargestellt. Der FD des Parents kann direkt über 'dd_parfd' angesprochen werden, ohne Umweg über den DD des Parents.
Jedes Directory hat ja einen ganz normalen Eintrag in seinem Parent. 'dd_parpos' bezeichnet nun die Dateiposition dieses eigenen Eintrags im Parent. Über 'dd_pardfd' und 'dd_parpos' kann also mit Hilfe von Fseek und Fwrite direkt der eigene Eintrag angesprochen und verändert werden. 'dd_lpos' zeigt auf die höchste Dateiposition, die bisher bei Suchoperationen in diesem Directory erreicht wurde. Dies ist für das automatische Erweitern des DD-Baumes wichtig (s. Teil 2).

```
typedef struct
{  char dd_name[11];          /* Directory-Name                               */
   int dd_stcl;               /* Start-Cluster                                */
   unsigned int dd_time;      /* Erstellungs-Zeit                             */
   unsigned int dd_date;      /* Erstellungs-Datum                            */
   FD *dd_dfd;                /* Zeiger auf eigenen FD                        */
   DD *dd_pardd;              /* Zeiger auf DD des parent directory           */
   DD *dd_chdd;               /* Zeiger auf DD eines child directory          */
   DD *dd_link;               /* Zeiger auf DD des nächsten childs            */
   DMD *dd_dmd;               /* Zeiger auf zugehörigen DMD                   */
   FD *dd_parfd;              /* Zeiger auf FD des parent directory           */
   long dd_parpos;            /* Position des eigenen Eintrags im parent      */
   long dd_lpos;              /* letzte Directory-Suchposition                */
   FD *dd_fdl;                /* Zeiger auf ersten User-Datei-FDi             */
} DD;
```

*Abb. 2: Directory Descriptor*

Damit können wir auch noch einmal auf die FDs von Anwenderdateien der letzten Folge zurückkommen. Die Verbindungen mit dem zugehörigen Directory-Descriptor sind in Abb. 5 dargestellt. Es ist praktisch eine vereinfachte Version von Abb. 4.
Hier noch ein Hinweis für diejenigen, die sich mit der internen Speicherverwaltung des GEMDOS auskennen. Die DDs benötigen vier 8-Wort-Einheiten. Freigegebene DDs werden daher ebenso wie die FDs in der 'mifl'-Liste 4 aufbewahrt.

## Initialisierung des DD-Baumes

Wenn ein Laufwerk zum ersten Mal angesprochen wird, richtet GEMDOS, wie Sie schon wissen, einen "Drive Media Descriptor" (DMD) ein. Dabei wird automatisch der DD für das Root Directory erzeugt. Damit ist ein Minimal-Baum vorhanden, der bei nachfolgenden Operationen erweitert wird.
Der Baum wird dabei im Laufe der Zeit immer nur vergrößert. GEMDOS hat quasi ein perfektes Gedächtnis, was die Existenz von Directories angeht. Eine Ausnahme bildet der Mediumwechsel, bei dem der gesamte DD-Baum aufgelöst wird. Hieraus ergeben sich auch Probleme mit der Harddisk. Da es hier i.allg. sehr viele Ordner und nie Medienwechsel gibt, sammeln sich recht viele DDs an, die leicht zu einer Überlastung der internen Speicherverwaltung führen können.

## Pfad-Handles - schon wieder neue Handles

Auf den DDs baut die Verwaltung der Standardpfade auf. Intern wird jeder Standardpfad über ein "Pfad-Handle" angesprochen. Diese Pfad-Handles sind für den Programmierer legal nicht zugänglich. Pfad-Handles liegen zwischen 1 und 39, hinzu kommt noch die Null als Dummy-Handle, das verwendet wird, wenn kein Pfad festgelegt ist.
Der Pfad selbst ist durch einen Zeiger auf den DD des Pfadendes, also des letzten Ordners des Pfades, eindeutig charakterisiert. Die Zuordnung dieser Zeiger zu den Pfad-Handles erfolgt über eine Tabelle ('pathx[]'), die zu den globalen GEMDOS-Variablen gehört (Tab. 1). 'pathx[5]' liefert also z.B. den Pfad für Pfad-Handle 5. Die Gültigkeit eines Elements von 'pathx[]' wird durch 'pthcntx[]' angezeigt.
Beim Starten von Tochterprozessen werden die Standardpfade mit ihren Handles "vererbt". Daher kann ein Handle mehrmals vergeben sein. Darüber wird mit 'pthcntx[]' (Tab. 1) Buch geführt. Für jedes Pfad-Handle steht hier die Anzahl der Prozesse, denen der jeweilige Standardpfad bekannt ist. Null zeigt an, daß das Pfad-Handle unbenutzt ist. Dann ist das zugehörige 'pathx[]' undefiniert (es muß nicht unbedingt NIL sein).
Nun muß GEMDOS noch wissen, welche Standardpfade jeder Prozeß hat. Dazu findet sich im Prozeßdescriptor (PD) das Feld 'char p_drvx[16]', in dem für jedes der möglichen 16 Laufwerke das Handle des Standardpfades steht (Null falls noch kein Standardpfad vergeben wurde, d.h. das Laufwerk noch nicht angesprochen wurde).

## Vererbung von Pfaden

Bei 'Pexec' werden alle Pfad-Handles aus 'p_drvx[]' des Parent-PDs in den Child-PD kopiert, wobei 'pthcntx[]' inkrementiert wird. Kurioserweise geschieht dies auch bei Null-Handles, die eigentlich anzeigen, daß kein Standardpfad existiert. Dies führt dazu, daß 'pthcntx[0]' unsinnige Werte enthält.
Dadurch wird erreicht, daß der Child-

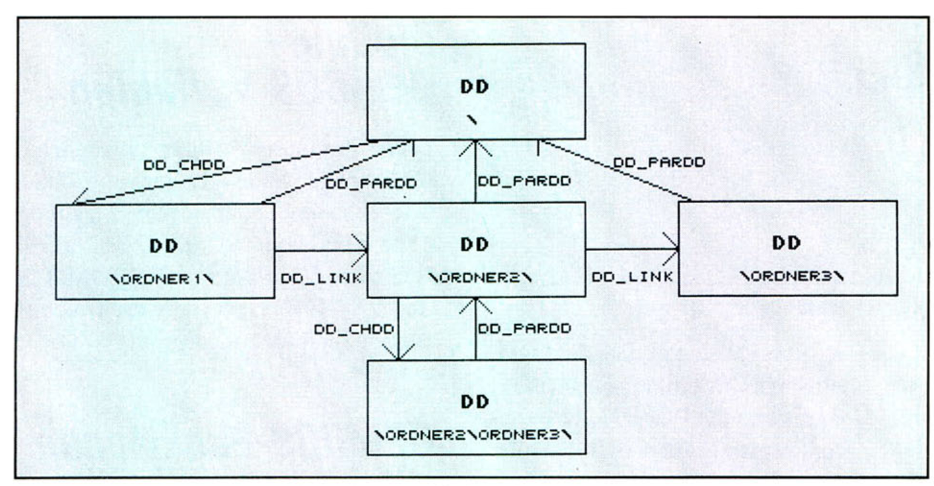



*Abb. 3: Baum der Directory Descriptoren (Beispiel)*

Prozeß die Pfade des Parents erhält, ohne daß für jeden Pfad ein neues Handle vergeben werden muß. Erst wenn das Child einen Pfad umsetzt, wird ein neues Handle fällig. Da sich an den Pfad-Handles des Parents nichts ändert, bleiben dessen Pfade von allen Aktionen des Childs unberührt.

Trotzdem sind 39 Pfad-Handles ein wenig knapp kalkuliert. Bei Verwendung wirklich aller 16 Laufwerke und Programme, die Pfade vieler Laufwerke neu setzen (z.B. Command Line Interpreter, Desktop) kann die Zahl gleichzeitig im Speicher haltbarer Prozesse schon auf 2-3 beschränkt sein.

Beim Terminieren eines Prozesses mit einer der 'Pterm'-Funktionen wird 'pthcntx[]' bei allen gültigen Handles dekrementiert. Hier werden Null-Handles extra abgefangen. Es wird jedoch nicht berücksichtigt, ob 'pthcntx[]' auch wirklich echt größer als Null ist. Dies dürfte eigentlich auch nicht vorkommen, aber auf Grund zweier Fehler (siehe weiter unten) kann dieser Fall doch eintreten, und schon hat man negative Werte.

Mit Abb. 6 ist ein Beispiel für die Pfadverwaltung mit zwei Prozessen gegeben. Prozeß 1 ist der Parent und hat Pfade für die Laufwerke A: und B: definiert. Prozeß 2 ist das Child und hat zunächst beide Pfade geerbt. Dann hat er einen Pfad für C: festgelegt und den Pfad für B: umdefiniert. Vergleichen Sie auch mit 'Dsetpath'.

So weit, so gut. Probleme tauchen aber auf, wenn man Programme mit 'Pexec' einmal laden, resident im Speicher halten und mehrmals starten möchte. Denn das Vererben der Pfade geschieht bei allen 'Pexec'-Modi außer 4 ("nur starten"). Die Übertragung der Pfade geschieht also nur beim ersten Start eines Programms und das auch noch zum Zeitpunkt des Ladens. Deshalb wäre es sinnvoller, dies mit den Modi 0 und 4 ("laden und starten" und "nur starten") zu verknüpfen.

Ähnliches gilt übrigens auch für die I/O-Umleitung. Beim 'Pterm' wäre es besser, wenn der PD wieder in den Zustand vor dem 'Pexec' gebracht, d.h. 'p_drvx[]' gelöscht würde. Während letzteres noch "von Hand" gemacht werden könnte, ist ersteres Problem nicht lösbar - halbwegs saubere Programmierung vorausgesetzt. Hier sei an die Atari-Programmierer appelliert, in Zukunft für Besserung zu sorgen.

## Anmeldung von Laufwerken

Nach der Initialisierung des GEMDOS beim Systemstart sind keinerlei Pfad-Handles belegt (alle 'p_drvx[]' sind gelöscht), d.h. es gibt für GEMDOS noch keine Standardpfade. Für den Anwender gilt jedoch das Root Directory als Standardpfad, solange kein anderer mit 'Dsetpath' ausgewählt wird. Daher sorgt GEMDOS dafür, daß bei einem Zugriff auf ein Laufwerk, das noch keinen Standardpfad hat, automatisch das Root Directory als Standardpfad gesetzt wird. Dies geschieht in der Routine 'd_chkdrv', die bei allen Laufwerks- und Pfadoperationen aufgerufen wird, bevor mit Pfad-Handles usw. gearbeitet wird.

'd_chkdrv' wurde schon in der April-Ausgabe unter "Anmeldung von Laufwerken" beschrieben, jedoch folgt hier noch einmal das wesentliche, um den Zusammenhang zur Pfadverwaltung besser herauszustellen.

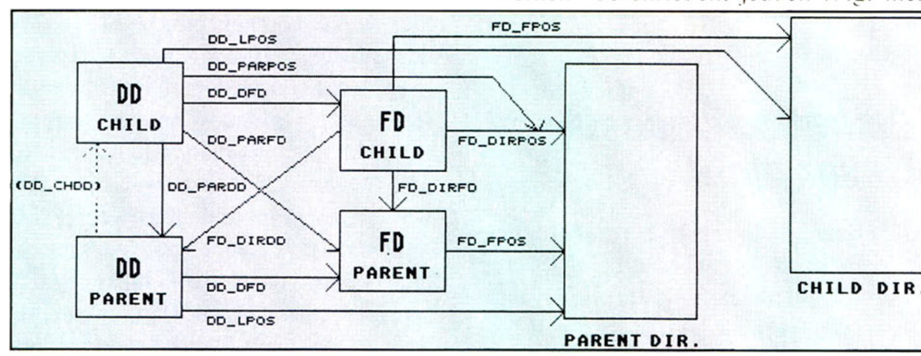



*Abb. 4: Child und Parent Directory*

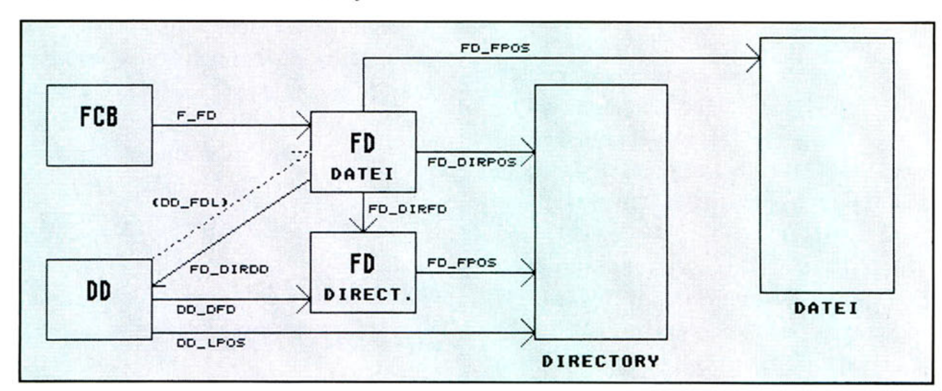



*Abb. 5: Anwenderdatei und Directory*

GEMDOS merkt sich, welche Laufwerke ihm bekannt sind, d.h. schon einmal angesprochen wurden. Bei bisher noch nicht bekanntem Laufwerk wird überprüft, ob es überhaupt ansprechbar ist (mit 'Getbpb'), wobei alle elementaren Strukturen für das Laufwerk (DMD, FD für FAT und Root Directory, DD des Root Directory) initialisiert werden. Existiert das Laufwerk nicht, bricht 'd_chkdrv' hier ab. Ansonsten ist das Laufwerk auf jeden Fall bekannt.

Ein Null-Pfad-Handle in 'p_drvx[]' zeigt an, daß noch kein Standardpfad festgelegt wurde. In diesem Fall wird ein freies Pfad-Handle ermittelt, der DD des Root Directory, der zu diesem Zeitpunkt auf jeden Fall vorhanden ist, in 'pathx[]' ein-

```
DD *pathx[40];      /*  $564c/  $564c/  $7dee: Standardpfade              */
char pthcntx[40]; /*  $58c4/  $58c4/  $8066: Zahl der Prozesse          */
long inv_pos        /* $17b06/$fd1c16/$fd3042: Konstante                  */
    = -1L;
char dirown[]       /* $17b0a/$fd1c1a/$fd3046: Directory-Eintrag '.'      */
    = { '.',0x20,0x20,0x20,0x20,0x20,
        0x20,0x20,0x20,0x20,0x20,
        0,0,0,0,0,0,0,0,0,0,0 };
char dirpar[]       /* $17b20/$fd1c30/$fd305c: Directory-Eintrag '..'     */
    = { '.','.',0x20,0x20,0x20,0x20,
        0x20,0x20,0x20,0x20,0x20,
        0,0,0,0,0,0,0,0,0,0,0 };
```

*Tabelle 1: Globale GEMDOS-Variablen zur Directory-Verwaltung*

getragen und 'pthcntx[]' auf Eins gesetzt. Damit ist das Root Directory als Standardpfad definiert. Das gleiche passiert auch, wenn das Pfad-Handle in 'p_drvx[]' zwar gültig ist, aber 'pathx[]' ein NIL-Zeiger ist. Dieser Fall soll der besonderen Behandlung von Medienwechseln dienen.

## Diskettenwechsel - lieber nicht

Wie innerhalb dieser Serie schon des öfteren erwähnt, müssen bei Medienwechseln, die beim ST i.allg. auf Disketten beschränkt sind, alle das Laufwerk betreffenden internen Datenstrukturen freigegeben und gegebenenfalls neu initialisiert werden. Dies gilt selbstverständlich auch für die Pfadverwaltung. Deshalb gibt der "Media change"-Handler sämtliche DDs und FDs frei. Außerdem werden alle Pfade des betroffenen Laufwerks ungültig gemacht, indem 'pthcntx[]' auf Null gesetzt wird. Außerdem wird 'pathx[]' auf NIL gesetzt, um die oben erwähnte Sonderbehandlung durch 'd_chkdrv' zu erzwingen, die aber nicht die rechte Wirkung zeigt. Das Problem besteht darin, daß die in den PDs gespeicherten Pfad-Handles von nun an ungültig gemachte Pfade bezeichnen.

Im Falle, daß der Pfad vorher nur vom eigenen Prozeß benutzt wurde, wird das Pfad-Handle bei der nächsten Gelegenheit neu vergeben, womöglich für den Standardpfad eines anderen Laufwerks. Dies hat dann zur Folge, daß dieser Pfad auch für das andere Laufwerk gilt. Und schon herrscht Chaos und der Programmierer wundert sich.

Abgesehen davon ist natürlich die Zählung mit 'pthcntx[]' durcheinandergeraten, was weitere "Spätfolgen" (z.B. negative Werte in 'pthcntx[]' nach der Prozeß-Terminierung!) nach sich ziehen kann.

Der Fehler ließe sich beseitigen, indem nur 'pathx[]' gelöscht würde, 'pthcntx[]' aber unverändert bliebe. Die Handles blieben dann belegt und die 'p_drvx[]'-Felder würden weiterhin gültige Handles enthalten.

In den meisten Fällen wird das Laufwerk mit einem neuen Medium ansprechbar sein, so daß dann die Pfad-Handles wieder gültig werden sollten. Die Routine 'd_chkdrv', die vor jeder Benutzung der Pfade zum Einsatz kommt, fragt diesen Fall (Laufwerk bekannt, handle gültig, aber 'pathx[]' NIL) sogar schon extra ab, reagiert aber falsch darauf. Bisher vergibt sie einfach ein neues Handle, dem das Root Directory als Pfad zugeordnet wird, und setzt es auch noch in 'p_drvx[]'. Dies würde dann funktionieren, wenn dabei 'pthcntx[]' des alten Handles dekrementiert würde, da es nun einem Prozeß weniger bekannt ist. Dies fehlt vermutlich deshalb, weil auf Grund des ersten Fehlers 'pthcntx[]' schon vom "Media change"-Handler gelöscht wird.

Günstiger wäre es noch, gar kein neues Handle zu vergeben, sondern einfach 'pathx[]' auf das Root Directory zu setzen. Damit wäre nach einem Mediumwechsel der alte Pfad einfach gegen das Root Directory ausgetauscht worden.

## Directory-Suchoperationen

Alle GEMDOS-Funktionen, die einen beliebigen GEMDOS-Dateinamen (möglicher Pfad + Name) als Parameter haben, rufen eine wichtige Routine auf, die aus dem Dateipfad den DD des zugehörigen Directories und den übrig bleibenden eigentlichen Dateinamen ermittelt. Diese Routine heißt bei mir 'd_getdir'.

Eine weitere wichtige Funktion ist 'd_srcname', die in einem schon ermittelten Directory nach bestimmten Dateien sucht. Beide Routinen werden in Teil 2 ausführlich erläutert. Entscheidend ist hier nur, daß bei beiden Funktionen alle Directories, die "auf dem Weg" liegen, automatisch in den DD-Baum eingefügt werden. Wenn übergeordnete Routinen mit Directories und Pfaden arbeiten, können sie sich darauf verlassen, daß alle benötigten Strukturen hiernach definiert sind.

## Globale GEMDOS-Variablen

Eine Zusammenstellung findet sich in Tab. 1. Die Adressen sind wie üblich TOS-abhängig und beziehen sich auf das RAM-TOS vom 6.2.1986, das ROM-TOS vom 6.2.1986 und das ROM-TOS vom 22.4.1987 (von links nach rechts). 'dirown' und 'dirpar' werden bei 'Dcreate' erläutert.

## GEMDOS-Funktionen (TRAP #1)

### Funktion $47 Dgetpath

*long Dgetpath(char *buf, int drive)*

Der aktuelle Standardpfad vom Laufwerk 'drive' (0 für Standard-Laufwerk, 1 für A:, 2 für B: usw.) wird ermittelt und in 'buf' abgelegt. Der String beinhaltet nicht die Laufwerkskennung, beginnt und endet nicht mit '\' und ist nullterminiert. Der Standardpfad 'A:\ORDNER1\ORDNER2\' wird also als 'ORDNER1\ORDNER2' geliefert.

Es gibt keine Größenbeschränkung von Seiten des GEMDOS; 'buf' muß groß genug bereitgestellt werden. Eine Größe von 128 Zeichen dürfte im allgemeinen ausreichen.

**Rückgabewerte:**

- -46L (EDRIVE) Laufwerk existiert nicht, interner Fehler (kein interner Speicher, keine Pfad-Handles mehr)
- 0L alles ok

**Arbeitsweise**

Zuerst wird eine eventuelle Angabe des Standardlaufwerks ausge-wertet, d.h., das gewünschte Laufwerk wird ermittelt. Wie üblich erfolgt der Aufruf von 'd_chkdrv'. Beginnend beim DD des Directories am Pfadende (erreicht durch 'p_drvx[]' und 'pathx[]', wird sich mittels der 'dd_pardd'-Zeiger rekursiv bis zum Root Directory hochgearbeitet. Beim Rücklauf der Rekursion werden die Directory-Namen ('dd_name') zum Pfad zusammengesetzt.

Stack-Überlauf-Probleme dürfte es hier nicht geben, da der für die Ausführung von GEMDOS-Funktionen eigens bereitgestellte Stack 3 kB groß ist.

Dies ist eine richtig hübsche Routine; bei aller Kritik muß man die GEMDOS-Programmierer auch einmal loben.

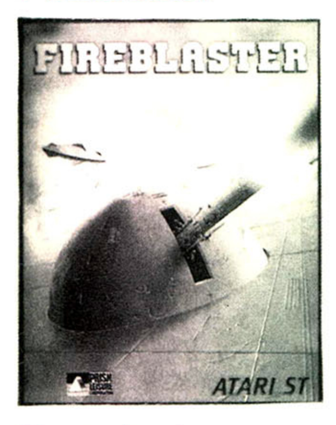
geht weiter...

# Drei Neuheiten auf einen Streich

## Pro Paket (drei Disketten — drei Spiele) nur **DM 79,—**



**1.**

| Fireblaster | Warzone | Space Station |
|---|---|---|
|  |  |  |
| Raumabenteuer | Weltraumkrieg | Spaceadventure |

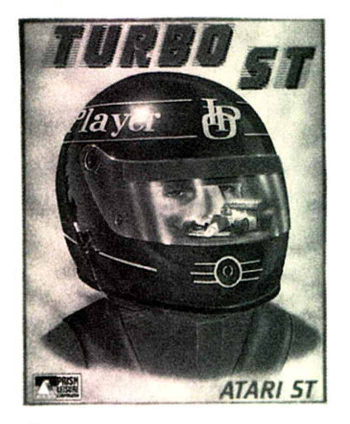
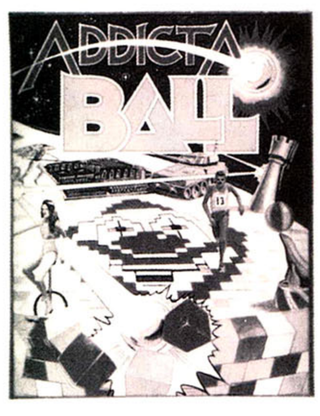
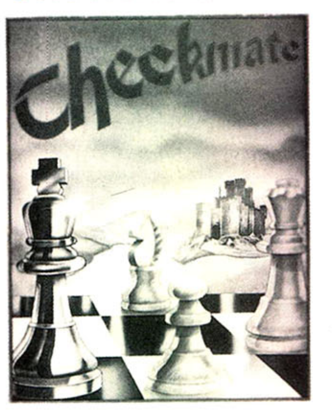


**DM 79,—**
(drei Disketten
drei Spiele)

**2.**



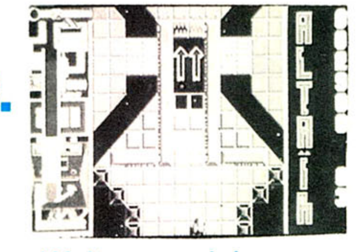
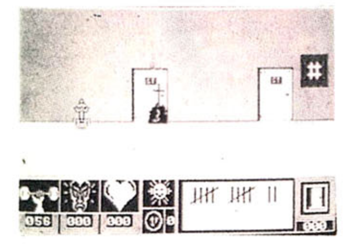
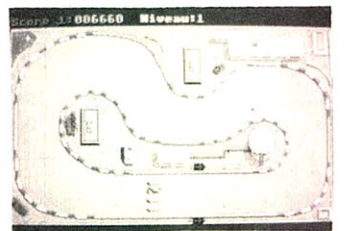


| Turbo ST | Addicta Ball | Checkmate |
|---|---|---|
|  |  |  |
| Autorennen | Weltraumabenteuer | Superschach |



**DM 79,—**
(drei Disketten
drei Spiele)

**3.**

| Altäir | Eden Blues | Turbo GT |
|---|---|---|
|  |  |  |
| Weltraumspiel | Actiongame | Autorallye |

**DM 79,—**
(drei Disketten
drei Spiele)

**Sofort bestellen — Anruf genügt — Mo bis Fr von 8.30—17.00 Uhr**
**Händleranfragen erwünscht!**

# Odin-Software GmbH


Hansmannstraße 19, D-5100 Aachen, Tel. (02 41) 55 00 07, Telefax (02 41) 55 21 49

## Funktion $3B Dsetpath

*long Dsetpath(char *path)*

Der Standardpfad für das in 'path' mit angegebene Laufwerk wird neu gesetzt. Er gilt nur für den eigenen Prozeß und wird von diesem an Tochterprozesse vererbt. 'path' endet nicht mit '\'.

Da Accessories immer unter dem PD der Haupt-Applikation laufen, können sich hier Konflikte ergeben. Wenn Accessories Pfade umsetzen, findet die Applikation eventuell ihre Dateien nicht mehr. Daher sollten Accessories 'Dsetpath' (und auch 'Dsetdrv') nicht benutzen, sondern jeweils komplette Pfade angeben.

**Rückgabewerte:**

-1..-31 BIOS-Fehlermeldung bei Diskzugriff

-1L (ERROR) Laufwerk existiert nicht, interner Fehler (keine Pfad-Handles)

-34L (EPTHNF) Pfad nicht gefunden, interner Fehler (keine Pfad-Handles, zu wenig interner Speicher)

-39L (ENSMEM) interner Fehler (zu wenig interner Speicher)

0L alles ok.

**Arbeitsweise**

Zuerst wird das Laufwerk, bei dem der Standardpfad geändert werden soll, ermittelt. Wenn kein Laufwerk explizit im Pfad angegeben ist, wird das aktuelle genommen.

Auch hier wird zunächst 'd_chkdrv' aufgerufen. Anschließend wird der alte Pfad ungültig gemacht, indem 'pthcntx[]' um eins erniedrigt wird. Wenn der Pfad von mehreren Prozessen benutzt wird, bleibt das Pfad-Handle also noch belegt.

Nun wird ein freies Pfad-Handle gesucht (erkannt an einer Null in 'pthcntx[]'), woraufhin mit 'd_getdir' der DD des gewünschten Directories (also des neuen Pfadendes) ermittelt wird.

Zuletzt wird ein Zeiger auf diesen DD in 'pathx[]' mit dem vorher gefundenen Pfad-Handle gespeichert, das zugehörige 'pthcntx[]' inkrementiert (hier also auf Eins gesetzt) und das Pfad-Handle in 'p_drvx[]' gemerkt.

Hier hat der Fehlerteufel mal wieder zugeschlagen:

Die Angabe des Laufwerks im Pfadnamen wird so gut wie ignoriert. Der Pfad bezieht sich immer auf das Standardlaufwerk. Beim zuerst ausgeführten 'd_chkdrv' wird noch das korrekte Laufwerk benutzt. 'd_getdir' wird aber ein Pfad ohne Kennung übergeben, so daß es auf dem Standardlaufwerk operiert. Dagegen verblaßt der andere Fehler geradezu.

Wenn kein Pfad-Handle mehr verfügbar ist (tritt praktisch nicht auf) oder der Pfad nicht gefunden wurde (durchaus möglich!), wird mit EPTHNF abgebrochen. Dabei ist aber der alte Pfad schon freigegeben ('pthcntx[]' dekrementiert), und die Referenz auf ihn über das Handle aus 'p_drvx[]' ist noch gültig. Dieser Fehler wirkt sich ganz ähnlich wie der unter "Diskettenwechsel" beschriebene aus.

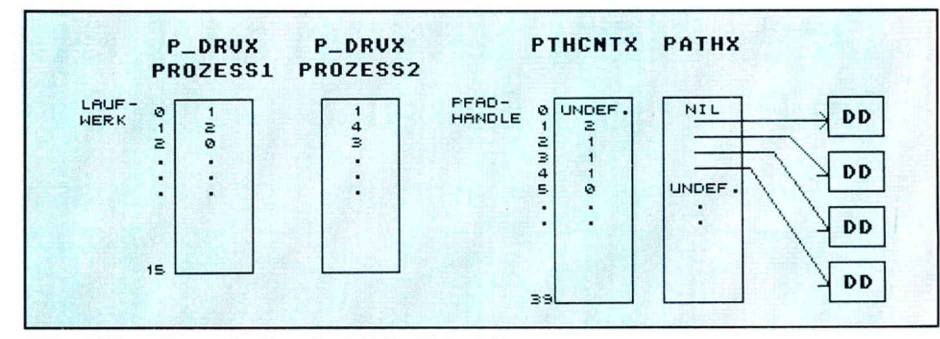



*Abb. 6: Verwaltung der Standardpfade (Beispiel)*

## Funktion $39 Dcreate

*long Dcreate(char *path)*

Ein neues Directory 'path' wird geschaffen. Eine Datei oder ein Directory gleichen Namens darf nicht schon existieren.

**Rückgabewerte:**

-1..-31 BIOS-Fehlermeldung bei Diskzugriff

-34L (EPTHNF) Pfad nicht gefunden, Laufwerk existiert nicht, illegaler Dateiname ('.' oder '..'), interner Fehler (keine Pfad-Handles, zu wenig interner Speicher)

-35L (ENHNDL) keine Datei-Handles mehr (intern wird ein Datei-Handle benötigt)

-36L (EACCDN) Diskette voll, Root Directory voll, Name existiert schon

-39L (ENSMEM) interner Fehler (zu wenig interner Speicher)

0L alles ok.

**Arbeitsweise**

'Dcreate' ist im Prinzip ein erweitertes 'Fcreate'. Daher wird zuerst das interne 'Fcreate' mit dem Dateiattribut $10 für Subdirectories aufgerufen. Zur Erinnerung sei hier noch einmal ein Teil der Beschreibung von 'Fcreate' wiederholt (s. letzte Folge).

Falls noch nicht vorhanden, wird für das Parent Directory des neu zu erzeugenden Directories ein FD angelegt. Die meisten Daten des FD korrespondieren mit entsprechenden des DD und werden direkt übernommen. Die Dateilänge ('fd_len') wird auf $7FFFFFFF gesetzt, da die Länge von Subdirectories nur durch die Kapazität des Mediums begrenzt ist. 'fd_link' und 'fd_mode' werden nicht benutzt.

Falls das Subdirectory schon existiert, wird abgebrochen. Nun wird im Parent Directory ein freier Eintrag gesucht und mit Dateiname, Attribut und timestamp initialisiert. Die Dateilänge und der Start-Cluster werden auf Null (letzteres zum Zeichen dafür, daß noch kein Cluster vergeben wurde) gesetzt. Außerdem wird dafür gesorgt, daß alle GEMDOS-Sektorpuffer zurückgeschrieben werden.

Jetzt wird die Datei mit einer der internen Open-Routinen, die auch das Handle generiert, richtig eröffnet (s. 'Fopen', letzte Folge). Hier werden Subdirectories wie jede andere Datei auch behandelt, d.h. das Subdirectory

bekommt einen ganz normalen Eintrag im Parent Directory.

Nach dem 'Fcreate' wird für das neue Subdirectory ein DD eingerichtet und in die Child-Liste des Parent Directories eingehängt. Außerdem wird auch gleich ein FD für das Subdirectory angelegt. Wenn für eine dieser Operationen nicht genügend interner Speicher vorhanden ist, wird zwar mit ENSMEM abgebrochen, doch die Subdirectory-Datei bleibt bestehen und geöffnet, insbesondere bleibt das Handle belegt und der Eintrag im Parent Directory bleibt erhalten.

Als nächstes ermittelt die FAT-Verwaltung einen freien Cluster und macht ihn zum ersten Cluster des Subdirectories. Bei vollem Speichermedium wird die Directory-Datei korrekt geschlossen, das Subdirectory im Parent Directory gelöscht und es erfolgt eine Fehlermeldung (EACCDN).

Nun werden die ersten zwei Einträge im Subdirectory ('.' und '..') generiert. Die Roheinträge sind in den globalen GEMDOS-Variablen zu finden (Tab. 1). Einziger Fehler hierbei: Der timestamp in diesen Einträgen wird nicht ins Intel-Format konvertiert. Wenn das Parent Directory das Root Directory ist, wird als Start-Cluster von '..' Null eingesetzt (an Stelle der GEMDOS-eigenen negativen Cluster-Nummer).

Zuletzt wird die Subdirectory-Datei korrekt geschlossen, wobei alle Sektorpuffer zurückgeschrieben werden.

## Funktion $3A Ddelete

*long Ddelete(char *path)*

Das Subdirectory 'path' wird gelöscht. Es muß dabei leer sein (bis auf die Einträge '.' und '..').

**Rückgabewerte:**

- -1..-31 BIOS-Fehlermeldung bei Diskzugriff
- -34L (EPTHNF) Pfad nicht gefunden, Laufwerk existiert nicht, interner Fehler (keine Pfad-Handles, zu wenig interner Speicher)
- -36L (EACCDN) Directory nicht leer
- -39L (ENSMEM) interner Fehler (zu wenig interner Speicher)
- -65L (EINTRN) interner Fehler (s. "Arbeitsweise")
- 0L alles ok

**Arbeitsweise**

Zuerst wird mit 'd_getdir' der DD des zu löschenden Subdirectories ermittelt und gegebenenfalls ein FD angelegt. Nun wird überprüft, ob das Subdirectory auch tatsächlich leer ist. Ab dem dritten Eintrag des Subdirectories werden alle Einträge, die mit $E5 beginnen (gelöschte Datei), überlesen. Wenn danach das Directory-Ende, erkannt an einem mit $00 beginnenden Eintrag oder am Ende der Directory-Datei, erreicht wird, ist das Subdirectory leer.

Als nächstes wird das Subdirectory in der Child-Liste seines Parents gesucht. Dies muß eigentlich immer klappen. Es gibt aber eine Sicherheitsabfrage, die EINTRN zurückgeben soll, wenn der DD des Subdirectory nicht auffindbar ist. Leider ist diese Abfrage falsch programmiert, so daß es stattdessen einen Bus Error (2 Bomben) gibt.

Noch zwei weitere Fälle, die eigentlich nicht auftreten dürfen, werden abgefangen und als EINTRN gemeldet: Es gibt noch offene Dateien im leeren Subdirectory ('dd_fdl' ungleich NIL), oder das Subdirectory hat selbst noch Subdirectories ('dd_chdd' ungleich NIL).

Jetzt wird der DD aus dem DD-Baum ausgehängt, DD und FD werden der internen Speicherverwaltung zurückgegeben und das Subdirectory wird mit einer internen Version von 'Fdelete' wie eine normale Datei gelöscht.

Haben Sie schon einmal versucht, das Root Directory zu löschen? Eigentlich überflüssig zu erwähnen, daß dies nicht abgefangen wird. 'Ddelete' bedankt sich beim Zugriff auf den Parent-DD mit zwei netten Bömbchen.

## Funktion $56 Frename

*long Frename(int dummy, char *oldpth, char *newpth)*

Die Datei 'oldpth' wird in 'newpth' umbenannt. 'newpth' muß das gleiche Laufwerk bezeichnen, darf jedoch ein anderes Directory spezifizieren. Die Datei kann also nicht zwischen zwei Laufwerken kopiert, aber innerhalb eines Laufwerks verschoben werden. Dabei werden nur Directory-Verweise geändert, die Datensektoren der Datei bleiben an ihrem Platz, so daß 'Frename' wesentlich schneller als Kopieren und anschließendes Löschen der Datei ist. Darum ist es besonders schade, daß man dies vom Desktop aus nicht ausnutzen kann.

'dummy' ist unbenutzt und sollte Null sein, um Kompatibilität zu späteren GEMDOS-Versionen zu wahren.

**Rückgabewerte:**

- -1..-31 BIOS-Fehlermeldung bei Diskzugriff
- -33L (EFILNF) Datei nicht gefunden, interner Fehler (zu wenig interner Speicher), illegaler Quelldateiname, Datei ist Subdirectory
- -34L (EPTHNF) Quell- oder Zielpfad nicht gefunden, Laufwerk existiert nicht, interner Fehler (keine Pfad-Handles, zu wenig interner Speicher)
- -35L (ENHNDL) kein Datei-Handle für Datei im Quellpfad
- -36L (EACCDN) Zieldatei existiert schon, Datei ist 'read only'
- -39L (ENSMEM) interner Fehler (zu wenig interner Speicher)
- -48L (ENSAME) unterschiedliche Laufwerke in Quell- und Zielpfad
- -65L (EINTRN) interner Fehler (s. "Arbeitsweise")
- 0L alles ok.

**Arbeitsweise**

Zuerst wird mit 'Fsfirst'/'Fsnext' (s. Teil 2) überprüft, ob es eine Datei gleichen Namens wie 'newpth' schon gibt. Dann werden mit 'd_getdir' die DDs von Quell- und Zielpfad bestimmt. Daraufhin werden Quell- und Ziellaufwerk miteinander verglichen, um evtl. mit ENSAME abzubrechen. Die Datei wird mit 'Fopen' und Modus 2 ("lesen und schreiben") eröffnet, daher erfolgt hier ein Abbruch bei 'read only'-Dateien. Wenn Quell- und Zieldirectory gleich sind, wird direkt mit internem 'Fwrite' der alte Dateiname im Directory durch den neuen ersetzt.

Bei verschiedenen Directories wird zuerst die Datei im Quelldi-rectory gelöscht, indem $E5 ins erste Zeichen des Dateinamens geschrieben wird. Dann werden Dateiattribut, Start-Cluster, Dateilänge und timestamp gemerkt. Jetzt erst wird mit 'Fcreate' und dem alten Dateiattribut ein neuer Dateieintrag im Zieldirectory geschaffen. Die gemerkten Daten werden ins Zieldirectory übertragen und Zieldatei und Zieldirectory geschlossen, so daß alle Änderungen hier auf das Speichermedium übertragen werden.

In beiden Fällen wird zum Schluß die Quelldatei geschlossen. Nur wenn hierbei kein Fehler auftritt, wird die Quelldirectory-Datei ebenfalls geschlossen. Beim Umbenennen zwischen Ordnern

ist vor allem die Reihenfolge der Aktionen zu beachten. Wenn beim Anlegen des neuen Eintrags etwas schiefgeht (z.B. BIOS-Fehler beim Zugriff auf Diskette), ist die Datei im Quelldirectory schon gelöscht!
Schlimmer noch, daß beim 'Fcreate' für die Zieldatei Fehler nicht erkannt werden, was zu Bombenhagel führen muß. Und zu diesem Zeitpunkt ist die Datei ebenfalls schon gelöscht...
Mögliche Fehler sind z.B. "Diskette voll", "Root Directory voll", "keine Datei-Handles mehr". Auch die Abfrage zu Beginn, ob der Zielname schon existiert, greift nicht immer. 'Fsfirst' wird nämlich mit dem Suchattribut Null aufgerufen. Deshalb werden "versteckte Dateien" ('hidden-Bit' oder 'system-Bit' im Dateiattribut gesetzt) nicht bemerkt. Das spätere 'Fcreate' löscht nun rücksichtslos die versteckte Datei, um den neuen Eintrag für die umzubenennende Datei zu schaffen.

Nächsten Monat geht es dann weiter mit Teil 2 der Directory-Verwaltung.

*Alex Esser*

*ENDE*

# WENN DER ATARI ST MIT DER GABI ANBANDELT



**Die Gabriele 9009 von Triumph Adler ist eine der meistverbreitetsten elektronischen Typenradschreibmaschinen. Leider kann sie nur über eine teure Schnittstellenbox als Computerdrucker verwendet werden. Dieser Artikel gibt einen Einblick in spezielle Druckerfunktionen, stellt das integrierte Schreibmaschineninterface vor und zeigt abschließend eine Softwarelösung auf, die einen direkten Anschluß der Schreibmaschine am ATARI ST doch ermöglicht.**

Von Triumph Adler wird eine Familie von Typenradschreibmaschinen angeboten, die sich zwar bezüglich der Schreibmaschinen-Funktionalität unterscheiden, die aber alle die gleiche integrierte Schnittstelle für einen Druckeranschluß bieten. Folgende Modelle gehören zu dieser Familie:

- Gabriele 9009
- Beta 8100
- SE 305
- SE 310
- SE 315
- SE 320
- SE 325

Außerdem besitzt noch das Schreibmaschinenmodell IBM Action Writer 6715, ein weitgehend mit der Gabriele 9009 baugleiches Modell, das gleiche Interface. Die nachfolgenden Ausführungen gelten für alle genannten Schreibmaschinen, auch wenn meist nur die Gabriele 9009 namentlich genannt wird.

Leider kann die in der Schreibmaschine integrierte Schnittstelle nicht zum direkten Anschluß an einen Computer verwendet werden, sondern es muß als "Dolmetscher" noch eine Interface-Box (TA-Bezeichnung IFD 1) zwischen Schreibmaschine und Computer geschaltet werden. Die Interface-Box bietet zum Computeranschluß eine parallele (Centronics-) Schnittstelle und eine serielle (V.24-) Schnittstelle. Trotz der vielen mit der Interfacebox realisierten Funktionen (z.B. Fettschrift oder Proportionalschrift) macht der hohe Preis der IFD1 eine Anschaffung gerade für den privaten Kunden nicht leicht, zumal sich die Kosten mit ca. DM 500,- im Bereich von 50% des Schreibmaschinenpreises bewegen. Diese Überlegung sowie der nicht ganz problemfreie Anschluß der IFD1 an den ATARI ST haben zur Entwicklung einer Softwarelösung geführt.

## Kleines "Schreibmaschinen-Einmaleins"

Bevor ich näher auf das Thema Kommunikation Computer - Schreibmaschine eingehe, möchte ich als Einleitung die Grundkomponenten einer elektronischen Typenradschreibmaschine am Beispiel der Gabriele 9009 vorstellen.

Die Schreibmaschine besteht im wesentlichen aus folgenden Modulen:

- **Tastaturleiterplatte** mit Tastern und LEDs
- **Logikkomponente** mit einem Mikroprozessorsystem, das sowohl die Druckersteuerung als auch die Funktionalität der Schreibmaschine (z.B. Funktionen "Automatische Korrektur", "Automatisches Unterstreichen" etc.) realisiert
- **Drucker** mit Schreibwalze und Druckschlitten

Die Funktionen des Druckers werden hierbei durch insgesamt vier Schrittmotoren umgesetzt:

- der **Schreibwalzenmotor** ist zuständig für die vertikale Bewegung des Papiers. Er wird beispielsweise dann gebraucht, wenn ein Linefeed den Papiervorschub auf eine neue Schreibzeile bringen soll. Ein Schritt des Motors bewirkt aufgrund der Getriebeübersetzung einen Vorschub um 1/96" (ca. 0,26 mm). Die weltweit standardisierte Grundzeile hat einen Abstand von 1/6" (ca. 4,23 mm); der Schrittmotor muß hierfür also um insgesamt 16 Schritte weitergeschaltet werden.

- der **Farb- und Korrekturbandmotor** steuert das Heben und den Transport der genannten Bänder. Auf diesen Motor hat man als Anwender keinen Zugriff, da er vom Betriebssystem der Schreibmaschine selbstständig entsprechend der gewünschten Aktion angesteuert wird.
- der **Schlittenmotor** sorgt für die (horizontale) Bewegung des Druckschlittens in beiden Richtungen, wobei ein Schritt des Motors eine Bewegung um 1/120" (ca. 0,21 mm) realisiert. Für die Ansteuerung unterscheidet man zwischen Bewegungen, die vom Betriebssystem automatisch nach dem Abdruck eines Zeichens ausgeführt werden und sogenannter direkter Bewegungsausführung (z.B. Carriage Return oder Tabulator).
- der **Typenradmotor** veranlaßt die Drehung des Typenrads. Aufgrund der Getriebeübersetzung entspricht ein Motorschritt der Drehung um eine Typenradspeiche. Alle von TA angebotenen Typenräder haben 100 Speichen, es stehen somit 100 verschiedene Zeichen zur Verfügung. Obwohl eine Vielzahl von Typenrädern mit verschiedenen Schriftarten verwendet werden kann, besitzen alle Typenräder der Ländervariante "deutsch" die gleiche Zuordnung der Zeichen zur Speichennummer.

Da die verschiedenen Zeichen unterschiedlich große Druckflächen besitzen (z.B. "." verglichen zu "W"), muß zur Erreichung eines gleichmäßigen Abdrucks jedem Zeichen eine spezifische Abschlagsenergie zugeordnet werden. Der Abschlag erfolgt durch einen mit konstanten Strom beaufschlagten Magneten, während die Abschlagsenergie durch Variierung der Bestromungszeit reguliert wird. Jedem Zeichen des Typenrades sind somit zwei Werte zugeordnet, nämlich die Speichennummer sowie die spezifische Abschlagszeit. Im normalen Schreibmaschinenbetrieb hat der Bediener mit all diesen Details nichts zu tun, sondern beschränkt sich aufs Betätigen der verschiedenen Tasten, während das Betriebssystem für die Umsetzung der oben genannten Funktionen sorgt. Wichtig werden die Informationen jedoch, wenn man wie wir die Schreibmaschine als direkt angesteuerten Drucker einsetzen möchte.

## Was kann das integrierte Schreibmaschineninterface?

Alle oben aufgeführten Schreibmaschinenmodelle besitzen an der Geräterückseite eine achtpolige DIN-Buchse. Auf diese Buchse herausgeführt ist kein Tonbandanschluß, sondern Versorgungsspannungen sowie Handshake- und Datenleitungen für eine serielle Schnittstelle. Diese Schnittstelle wird je nach Modell entweder direkt vom Mikroprozessor realisiert oder es wird ein USART-Baustein verwendet. Da die Signale jedoch nicht über entsprechende Treiberbausteine geführt wurden, stehen nur TTL-Pegel (0V —> "Low" und +5V —> "High") und keine V.24-Pegel (+12V —> "Low" und -12V —> "High")zur Verfügung. Will man einen Anschluß an einen Computer vornehmen, muß also unbedingt eine Signalumsetzung durchgeführt werden.
Eine Beschreibung der an der Buchse anliegenden Signale kann dem Bild 1 entnommen werden.

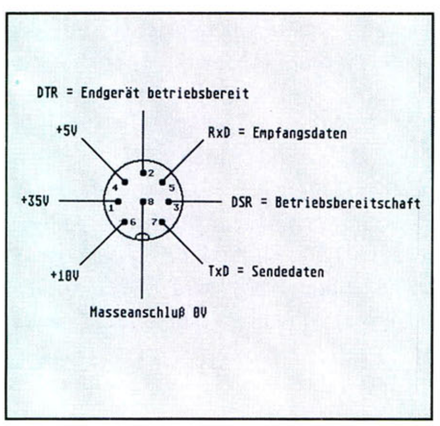



*Bild 1: Schnittstellenbuchse in Schreibmaschinenrückwand (Draufsicht)*

Man hat bei TA aber nicht nur an Treiberbausteinen gespart, sondern, was die Sache weit komplizierter macht, auch an einer Treibersoftware. Es besteht nämlich keine Möglichkeit, über die integrierte Schnittstelle einfach Zeichen im ASCII-Format zu schicken und zu hoffen, daß die Schreibmaschine diese entsprechend den eingestellten Parametern (Zeichen- und Zeilenabstand) ausdruckt. Vielmehr versteht der Schreibmaschinen-Mikroprozessor nur druckerspezifische Anweisungen im Interncode (z.B. "drehe Typenrad zur Speiche 6" oder "bewege den Schlitten um 10 Schritte nach rechts").
Aus diesem Grund sitzt in der Original-Interface-Box von TA eine Elektronikplatine mit eigenem Mikroprozessor, RAM- und EPROM-Speicher sowie einem weiteren USART-Baustein. Das von TA realisierte Konzept ermöglicht mit der IFD1 eine von der Schreibmaschine unhabhängige hohe Druckerfunktionalität, führt jedoch aufgrund des hohen Hardwareaufwands zu kaum vertretbaren Kosten.
Die Idee eines softwaremäßigen "IFD-Emulator" geht nun davon aus, die Intelligenz der IFD1 im Computer (hier ATARI ST) durch ein Programm nachzubilden, das beim Bootvorgang im Speicher fest installiert wird.

## Hardware-Anschluß

Wie oben bereits erwähnt, stehen bei den Schreibmaschinen alle benötigten Signale an der achtpoligen DIN-Buchse auf der Geräterückseite zur Verfügung.
Obwohl der Datentransfer hauptsächlich in der Richtung Computer zur Schreibmaschine abläuft, ist es erforderlich, die Verbindung bidirektional auszulegen. Der Grund liegt darin, daß die Schreibmaschine auch diverse Meldungen an den Computer schicken muß (z.B. "Online-Taste wurde gedrückt").
Die logische Verbindung wird durch eine Verdrahtung nach Tabelle 1 hergestellt (siehe unten).
Durch die Betriebssystem-Software wird das Datenformat der Schreibmaschinenschnittstelle auf folgende Parameter festgelegt:

- asynchrone Datenübertragung
- Übertragungsrate: 4800 Baud
- ein Startbit
- acht Datenbits
- ein Stopbit
- kein Paritätsbit.

| ATARI ST | Schreibmaschine (SM) | Erläuterung |
|---|---|---|
| RxD (Pin 3) | TxD (Pin 7) | Daten SM-->ST |
| TxD (Pin 2) | RxD (Pin 5) | Daten ST-->SM |
| RTS (Pin 4) | DSR (Pin 3) | Steuer ST-->SM |
| CTS (Pin 5) | DTR (Pin 2) | Steuer SM-->ST |

*Tabelle 1: So muß verdrahtet werden*

Doch bevor Sie jetzt beginnen, ein entsprechendes Kabel zu löten, komme ich noch zu dem oben bereits erwähnten Problem der unterschiedlichen Signalpegel.

Würde man die Komponenten ohne Umsetzer verbinden, wäre keine vernünftige Kommunikation möglich, außerdem könnten die Schnittstellentreiber beider Systeme zerstört werden. Das Problem muß dadurch gelöst werden, daß die Schreibmaschinensignale von TTL-Pegel auf V.24-Level umgesetzt werden.

V.24-Level bedeutet für
- High-Signal: - 3 V bis - 15 V
- Low-Signal: + 3 V bis + 15 V.

Entsprechend der Norm ist der Bereich zwischen -3 V und +3 V undefiniert.

Häufig findet man gerade im Hobbybereich unsymmetrische Anpaß-Schaltungen, die die Umsetzung einfach mit Hilfe eines Inverterbausteins realisieren.

Ein "Low" am Eingang des Inverters erzeugt am Ausgang ein Signal von ca. +12 V, was somit der Normanforderung genügt. Ein "High" am Eingang ergibt am Ausgang jedoch anstelle der geforderten negativen Ausgangsspannung nur eine Spannung von ca. 0 V. Obwohl dieser Pegel somit im undefinierten Bereich liegt, führt diese Umsetzung oft zum Erfolg.

Aufgrund mangelnder Störsicherheit und großer Abhängigkeit von Bauteiletoleranzen wird jedoch eine Anpaß-Schaltung auf "echte" V.24-Pegel empfohlen. Hierzu werden am Elektronikmarkt spezielle Treiberbausteine 75188 (bzw. 1488) und 75189 (bzw. 1489) angeboten, mit denen sich die geforderte Umwandlung einfach realisieren läßt. Da die genannten Bausteine diese Aufgabe in der gesamten PC-Welt (übrigens auch im ATARI-ST) erfüllen, handelt es sich hier um eine sehr kostengünstige Lösung. Allerdings hat die Sache einen nicht unerheblichen Haken: Für die Umsetzung der Signalpegel benötigt der Baustein 74188 als Versorgungsspannung neben + 12V auch die Spannung -12V.

Die Schreibmaschine stellt an der DIN-Buchse zwar verschiedene Versorgungsspannungen zur Verfügung, jedoch handelt es sich mit +5V, +10V und +35V nur um positive Gleichspannungen. Als Lösung bietet sich die Versorgung unserer kleinen Anpaß-Schaltung mit einem getrennten Netzteil an. Wer kein solches zur Verfügung stehen hat, oder sich nicht mit dem Gedanken an noch mehr Netzteile/Kabel auf dem Schreibtisch anfreunden kann und auch keine "Kunstschaltung"

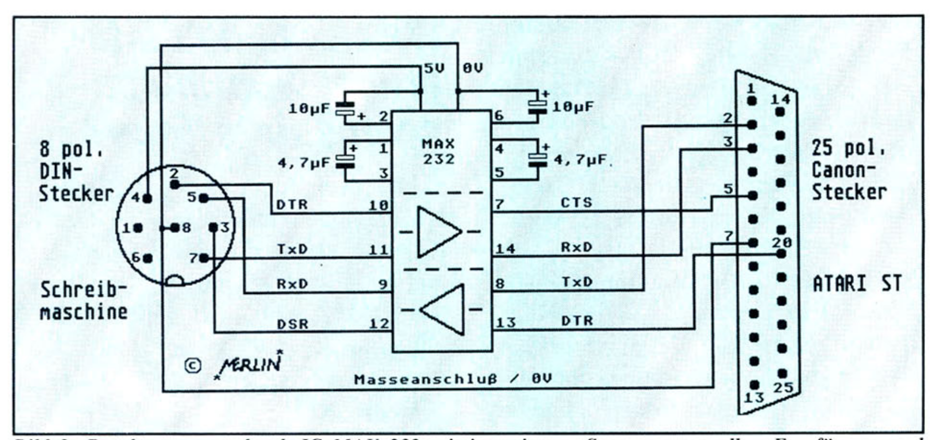



*Bild 2: Pegelanpassung durch IC MAX 232 mit integriertem Spannungswandler, Empfänger- und Treiberstufe*

zur Erzeugung der +/- 12V aus den vorhandenen Gleichspannungen diskret aufbauen will, dem sei der in Bild 2 dargestellte Schaltungsvorschlag empfohlen.

Hierzu folgende kurze Erläuterung:

Kern der Schaltung ist ein relativ neuer integrierter Schaltkreis mit der Bezeichnung MAX 232. Hierbei hat man es geschafft, auf einem Chip zwei Empfänger (Umwandlung +/-12V-Pegel auf 5V-Pegel) und zwei Sender (Umwandlung 5V-Pegel auf +/-12V-Pegel) zu vereinen. Der Clou ist aber, daß der Chip außerdem noch eine Schaltung integriert hat, die aus der Versorgungsspannung von +5V die oben angesprochenen Spannungen von -10V und +10V selbständig erzeugt, wobei als externe Beschaltung lediglich vier Kondensatoren benötigt werden.

Obwohl der Baustein MAX 232 mit einem Preis von ca. DM 13,- teurer ist als die genannten 75188 und 75189, bin ich doch der Meinung, daß dieser kleine Nachteil durch den wesentlich einfacheren und kompakteren Aufbau mehr als ausgeglichen wird.

Die oben vorgestellte Anpaß-Schaltung kann auf eine kleine Lochrasterplatine aufgebaut werden. Wenn man dabei geschickt ist, kann man den gesamten Aufbau in das Gehäuse des ST-Steckers unterbringen. Deshalb sollten unbedingt Tantal-Elkos verwendet werden.

## Betriebszustände

Hiermit sind die Steuerkommandos bzw. Meldewörter gemeint, die von der Schreibmaschine verstanden werden und entsprechende Aktionen auslösen.

Wir unterscheiden bei unserer Anwendung grundsätzlich folgende Zustände der Schreibmaschine:

OFFLINE: Die Schreibmaschine ist "Schreibmaschine", die auf der Tastatur vorhandene Online-Anzeige (LED) ist "aus", der Computer kann keine Druckbefehle schicken.

ONLINE: Die Schreibmaschine ist "Drucker", die Online-Anzeige (LED) ist "ein", der Computer kann Druckbefehle schicken, Eingabe an der Schreibmaschinentastatur ist nicht mehr möglich.

Da die Schreibmaschine prinzipiell nur auf Anweisungen von außen reagiert, muß bei Direktanschluß vom Computer die gesamte Steuerungsfunktionalität ausgeführt werden, so auch z.B. das Umschalten vom Zustand OFFLINE in den Zustand ONLINE. Hierzu müssen nachfolgend aufgeführte Kommandos verwendet werden, die alle eine Länge von zwei Bytes haben, wobei das erste Byte den Befehl definiert und das zweite Byte für eine Versionsnummer reserviert ist (üblicherweise Version 00H):

- **A0H 00H**: **CLEAR**-Kommando —> die Schreibmaschine geht in den OFFLINE-Zustand und wird wieder "Schreibmaschine".
- **A1H 00H**: **START**-Kommando —> Übergang zu ONLINE wird vorbereitet
- **A2H 00H**: **STX**-Kommando —> Schreibmaschine wird "Drucker", Zustand "ONLINE", Übertragung von Druckbefehlen kann beginnen
- **A3H 00H**: **ETX**-Kommando —> Datenübertragung wird unterbrochen und auf Fortsetzung gewartet




Dr. R. Witzmann

TEXTVERARBEITUNG
AUF DEM ATARI ST

**1ST WORD PLUS**

VERSTEHEN · BENUTZEN
DRUCKEN

MIT GROSSEM ÜBUNGSTEIL
1ST WORD TRAINER

Heim Verlag




# Endlich! Das Buch zu 1ST Word Plus mit großem Übungsteil 1ST Word Trainer

Die neuartige Einführung in die Textverarbeitung, die verspricht ein Klassiker zu werden. Ein Buch, das man auch einmal am Wochenende lesen möchte – begeisternd, und manchmal auch bissig – über „ein Programm", das nie ganz ausgereizt wurde.

**WICHTIGE MERKMALE:**

- Voll von Information, die menschlich verpackt ist
- Zeigt minutiös die Technik der Textverarbeitung, ohne ihre Faszination zu vergessen
- Vom Praktiker für Praktiker: Für Autoren und Ärzte... Sekretärinnen und Schwiegermütter... Journalisten und Geschäftsleute
- Für Anfänger, die schnell lernen wollen und nichts vom Computer verstehen
- Für Fortgeschrittene, die sich darüber hinaus für RAM-Disk, Zeitsparen, Grafik und Drucklegung interessieren
- Für Kenner, die mal sehen wollen, ob es bei der Textverarbeitung noch etwas gibt, das sie nicht wissen
- Mit vielen Beispielen und Grafiken auf über 300 Seiten

**AUS DEM INHALT:**

- Wie man sofort Text auf den Bildschirm bringt
- Der unendliche Zeichensatz und 1st Proportional
- Alle Menüs auf einen Blick
- Tippen, Korrigieren, Zeit sparen, Wörterbuch ...
- Formatieren und Umbruch
- Einbinden von Grafik, DEGAS im Text, selber malen, drucken
- Ist eine Art DTP möglich?
- 1ST MAIL – auf 30 Seiten so erklärt, daß es jeder versteht
- Zum ersten Mal: Vom Manuskript zum Buchdruck!
- Was man nicht im Handbuch findet: 100 Tips u. Tricks
- Verschiedene Bildgrößen
- Wie man den Treiber zum Großschreiben bringt
- „Was tun wenn...?" Eine lange Liste von Ratschlägen
- Der Umgang mit dem Drucker – Dichte, Schnelligkeit, Zeichensatz, Papiereinzug, Berechnung von Kopf- und Fußzeile
- Übersetzung der wichtigsten Fachausdrücke
- „1ST WORD TRAINER" – Aufgaben, Probleme, Fragen: Ein Übungsteil, der zum Denken auffordert.

Über 300 Seiten

**DM 49,–**

Unverbindlich empfohlener Verkaufspreise

**BESTELLCOUPON**

an Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Bitte senden Sie mir ________ Stück **1ST WORD PLUS** z. Preis von DM 49,– je Stück zzgl. DM 5,– Versandkosten unabhängig von der Bestellmenge. ☐ per Nachnahme ☐ Scheck liegt bei ☐ per Vorausüberw.

Name, Vorname ____________________ Straße, Nr. ____________________

PLZ, Ort ____________________ Unterschrift ____________________

Benutzen Sie auch die im ST-COMPUTER vorhandene Bestellkarte

**Heim Verlag**
Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

- **A4**H **00**H: **ENQ**-Kommando —> Schreibmaschine soll Zustand melden

Durch Senden dieser Kommandos kann man nun vom Computer aus zwischen den verschiedenen Zuständen umschalten, wobei die Reihenfolge der verschiedenen Befehle nicht willkürlich festgelegt werden kann, sondern genau eingehalten werden muß:

- Umschalten OFFLINE —> ONLINE:
  A0H 00H
  A1H 00H
  A4H 00H
  A2H 00H
- Umschalten ONLINE —> OFFLINE:
  A3H 00H
  A0H 00H.

Je nach Schreibmaschinentyp und Ausführungsstand befindet sich auf der Schreibmaschinentastatur eine spezielle Taste mit der Aufschrift "ONL" sowie zugehöriger LED. Bei anderen Varianten (z.B. SE 325 oder Gabriele 9009 neuesten Ausführungsstandes) wurde die Taste zwar weggelassen, ihre Funktion jedoch durch Kombination anderer Tasten nachgebildet (vergleichen Sie hierzu bitte die Bedienungsanleitung der Schreibmaschine).

Funktionell macht diese Taste (oder die entsprechende Tastenkombination) nichts anderes, als die Schreibmaschine zu veranlassen, einen entsprechenden Tastencode (ONL-Taste = 01H) an die Interfacebox bzw. in unserem Fall an den Computer zu schicken. Es ist dann am Gegenüber, die Schreibmaschine durch obengenannte Befehlssequenz auf ONLINE zu schalten. Für das Zurückschalten in den OFFLINE-Zustand von der Schreibmaschinentastatur aus gibt es ebenfalls eine Tastenkombination, die je nach Schreibmaschinentyp unterschiedlich ist (z.B. CE-Taste bei SE325, MOD+ONL bei "alten" Gabriele 9009 und SE 310). Auch hier sei auf die entsprechenden Bedienungsanleitungen verwiesen.

Wie wir gesehen haben, muß die Umschaltung zwischen den verschiedenen Betriebszuständen stets vom Computer aus gesteuert werden. Dies kann im einfachsten Fall dadurch geschehen, daß man ein Programm startet, das nach der Konfiguration der Schnittstellenparameter (Baudrate etc.) einfach die obengenannte Befehlssequenz zur Schreibmaschine schickt. Komfortabler kann man es gestalten, wenn man eine Interruptroutine im Speicher des ATARI ST installiert, die dann angesprungen wird, wenn die Schreibmaschine eine Meldung gesendet hat. Nach Dekodierung dieser Meldung wird entweder in das ONLINE- oder das OFFLINE-Programm verzweigt. Dieser Trick erlaubt es, den Betriebszustand quasi an der Schreibmaschine einzustellen, obwohl die eigentliche Steuerintelligenz vom Computer ausgeht.

## Druckerspezifische Codes

Wenn man die Schreibmaschine nun endlich in den ONLINE-Zustand gebracht hat, steht sie als Drucker zur Verfügung. Wie bereits einleitend erwähnt, ist es allerdings nicht möglich, jetzt die zu druckenden Zeichen einfach in ASCII-Code zu schicken. Vielmehr versteht die Schreibmaschine nur Befehlscodes, die sehr druckernah aufgebaut sind, wobei auch hier jeder Befehl aus zwei zusammengehörenden Bytes besteht. Nachfolgend möchte ich die verschiedenen Codes zusammenfassend auflisten:

- **direkte Bewegungsausführung** entweder horizontal durch Bewegung des Schlittens (Carriage Return) oder vertikal durch Drehen der Schreibwalze (Linefeed).

1. BYTE
Bit 0 —> )
Bit 1 —> )> höherwertiges Byte für Angabe der Bewegungsstrecke
Bit 2 —> )> in 1/120"- bzw. 1/96"-Schritten
Bit 3 —> )
Bit 4 —> "0" für Horizontalbewegung, "1" für Vertikalbewegung
Bit 5 —> "0" für rechts/vorwärts, "1" für links/rückwärts
Bit 6 —> "1" für Befehlscode
Bit 7 —> "1" für Befehlscode

2. BYTE
Bit 0 —> )
Bit 1 --> )
Bit 2 —> )
Bit 3 —> )> niederwertiges Byte für Angabe der Bewegungsstrecke
Bit 4 —> )> in 1/120"- bzw. 1/96"-Schritten
Bit 5 —> )
Bit 6 —> )
Bit 7 —> )

- **Grundstellung/Reset** veranlaßt einen Grundstellungslauf der verschiedenen Schrittmotoren (z.B. der Druckschlitten bewegt sich auf Grundposition-Nullgrad am linken Maschinenrand).

1. BYTE
Bit 0 —> "0"
Bit 1 —> "1"
Bit 2 —> "0"
Bit 3 —> "0"
Bit 4 —> "0"
Bit 5 —> "0"
Bit 6 —> "0"
Bit 7 —> "1"

2. BYTE
Bit 0 —> "1"
Bit 1 —> "1" für Grundstellungslauf des Schlittenmotors
Bit 2 —> "1" für Grundstellungslauf des Typenradmotors
Bit 3 —> "1" für Grundstellungslauf des Farbbandmotors
Bit 4 —> "0"
Bit 5 —> "0"
Bit 6 —> "0"
Bit 7 —> "0"

- **Horizontalschrittdefinition** legt die Anzahl von 1/120"-Schritten fest, die der Druckschlitten jedesmal nach dem Zeichenausdruck ausführt (Schreibteilung)

1. BYTE
Bit 0 —> "0"
Bit 1 —> "0"
Bit 2 —> "0"
Bit 3 —> "0"
Bit 4 —> "0"
Bit 5 —> "0"
Bit 6 —> "0"
Bit 7 —> "1"

2. BYTE Bit 0 —> )
Bit 1 —> )
Bit 2 —> )
Bit 3 —> )> Übergabe der gewünschten Schrittzahl (zwischen
Bit 4 —> )> 0 und 255 Schritten einstellbar)
Bit 5 —> )
Bit 6 —> )
Bit 7 —> )

- **Schreibzeichen** legt das zu druckende Zeichen fest. Definiert werden müssen hierbei der Typenradcode sowie die spezifische Abschlagsstärke.

1. BYTE Bit 0 —> )
Bit 1 —> )
Bit 2 —> )
Bit 3 —> )> Typenradcode (Speichennummer)
Bit 4 —> )
Bit 5 —> )
Bit 6 —> )
Bit 7 —> "0"

2. BYTE Bit 0 —> )
Bit 1 —> )
Bit 2 —> )
Bit 3 —> )> Abschlagsstärke
Bit 4 —> )
Bit 5 —> )
Bit 6 —> "0" für Bewegung nach rechts, "1" für Bewegung nach links
Bit 7 —> "0" für Unterdrückung der anschließenden Horizontalbewegung

- **Space (Leerschritt)** erlaubt eine "kleine" Schlittenbewegung nach rechts. Wird als Schrittzahl der Wert "0" übergeben, erfolgt eine Bewegung entsprechend der eingestellten Schreibteilung.

1. BYTE Bit 0 —> "1"
Bit 1 —> "1"
Bit 2 —> "0"
Bit 3 —> "0"
Bit 4 —> "0"
Bit 5 —> "0"
Bit 6 —> "0"
Bit 7 —> "1"

2. BYTE Bit 0 —> )
Bit 1 —> )
Bit 2 —> )
Bit 3 —> )> Übergabe der gewünschten Anzahl
Bit 4 —> )> von 1/120"-Schritten
Bit 5 —> )
Bit 6 —> )
Bit 7 —> )

- **Backspace** bildet das Gegenstück zu Space und veranlaßt eine Schlittenbewegung nach links

1. BYTE Bit 0 —> "0"
Bit 1 —> "0"
Bit 2 —> "1"
Bit 3 —> "0"
Bit 4 —> "0"
Bit 5 —> "0"
Bit 6 --> "0"
Bit 7 —> "1"

2. BYTE Bit 0 —> )
Bit 1 —> )
Bit 2 —> )
Bit 3 —> )> Übergabe der gewünschten Anzahl
Bit 4 —> )> von 1/120"-Schritten
Bit 5 —> )
Bit 6 —> )
Bit 7 —> )

Mit den oben aufgelisteten Befehls-Sequenzen kann die Schreibmaschine alle denkbaren Druckerfunktionen ausführen. Zu beachten ist jedoch, daß alle "Intelligenz" vom Computer ausgehen muß und die Schreibmaschine sich auf die Ausführung der empfangenen Druckbefehle beschränkt. Der Computer muß sich beim "normalen" Drucken beispielsweise merken, auf welcher Position der Druckschlitten steht, damit dieser bei einem Carriage-Return mit der richtigen Anzahl von Schritten nach links bewegt wird. Lediglich für die Horizontalbewegung nach dem Abschlag eines Zeichens besitzt die Schreibmaschine eine gewisse Selbständigkeit: Man definiert durch Angabe der Variablen "X" in der Befehlssequenz 83H + X die Anzahl der 1/120"-Schritte, die die Schreibmaschine dann nach jedem Zeichenabdruck selbständig ausführt und legt somit die Schreibteilung fest (X=12 —> Teilung 10/Elite oder X=10 —> Teilung 12/Pica).
Die an der Schreibmaschine eingestellten und durch LEDs auf der Tastatur angezeigten Parameter wie Zeilenabstand, Schreibteilung und Abschlagsstärke sind im Druckerbetrieb unwirksam.

## Zwischenbilanz

Mit den aufgeführten Daten bzw. Informationen ist es nun möglich, die Schreibmaschine in den Zustand ONLINE zu schalten, und über die entsprechenden Schrittmotoren alle denkbaren Aktionen zu realisieren. Um die graue Theorie mit etwas praktischem Leben zu erfüllen, habe ich ein kleines Demonstrationsprogramm in GFA-BASIC erstellt, das die bisher besprochenen Funktionen in sinnvoller Kombination einsetzt. Das Programm ist als Anregung für weitergehende Experimente gedacht und kann natürlich beliebig ergänzt und modifiziert werden. Da man über die Schreibwalze eine vertikale und über den Druckschlitten eine horizontale Bewegung steuern kann, wäre es beispielsweise denkbar, durch Befestigung eines Stiftes am Druckkopf einen Plotter nachzubilden. Der Stift kann dann in 1/120 " bzw. 1/96 " Schritten frei über das Papier bewegt werden.
In der nächsten Ausgabe werden wir dann ein Programm vorstellen, das die heute vorgestellten Befehlssequenzen zu einer leistungsfähigen Emulation der TA-Interfacebox zusammenfügt.

*Wilfried Keller*

```
  1:  ' ****************************************************
  2:  ' * DEMOPROGRAMM für Ansteuerung der Gabriele 9009   *
  3:  ' * über serielle Schnittstelle                      *
  4:  ' *                                                  *
  5:  ' * Copyright  ST-Computer - Ausgabe 7-1988          *
  6:  ' ****************************************************
  7:  '
  8:  ' Zuerst alle störenden V.24-Interrupts ausschalten
  9:  Dummy=Xbios(26,1)
 10:  Dummy=Xbios(26,10)
 11:  Dummy=Xbios(26,12)
 12:  Dummy=Xbios(26,9)
 13:  Dummy=Xbios(26,2)
 14:  Dummy=Xbios(26,11)
 15:  ' Schnittstelle auf 4800 Baud, 1 Start- und 1 Stopbit
 16:  ' 8 Datenbits und kein Parity-Bit einstellen !!
 17:  ' Dummy=Xbios(15,2,2,-1,-1,-1,-1)
 18:  '
 19:  ' Schreibmaschine auf Online schalten
 20:  Gosub Online
 21:  Print "Die Maschine ist ONLINE"
 22:  ' auf Eingabe warten
 23:  Print
 24:  Input "Bitte Return-Taste drücken ",A$
 25:  ' 500 Einzel-Schritte zu 1/120 inch nach rechts fahren
 26:  For R=1 To 500
 27:    Gosub Micro_step
 28:    Print "Die Maschine steht  jetzt auf Position Nr. ";R
                                                         ;" !!"
 29:  Next R
 30:  '
 31:  Print
 32:  ' 500 Einzel-Schritte zu 1/120 inch nach links fahren
 33:  For R=500 To 200 Step -1
 34:    Zeichen=&H84
 35:    Gosub Schr
 36:    Zeichen=1
 37:    Gosub Schr
 38:    Print "Die Maschine steht  jetzt auf Position Nr. ";R
                                                         ;" !!"
 39:  Next R
 40:  Print
 41:  '
 42:  ' Jetzt ein paar direkte Bewegungsausführungen
 43:  ' Zuerst 1000 Schritte (1/120 inch) nach rechts als
      Absolutbewegung
 44:  ' Grundcode für horiz. Bewegung nach rechts in Zeichen
 45:  Zeichen=&HC0
 46:  Schrittzahl=1000
 47:  Gosub Dir_bewegung
 48:  Print "Ich habe den Schlitten jetzt um 1000/120 Zoll
             nach rechts bewegt!"
 49:  Print
 50:  Gosub Delay3
 51:  '
 52:  ' jetzt 750 Schritt (1/96 inch) nach unten als
      Absolutbewegung
 53:  ' Grundcode für vert. Bewegung nach unten in Zeichen
 54:  Zeichen=&HD0
 55:  Schrittzahl=750
 56:  Gosub Dir_bewegung
 57:  Print "Ich habe das Papier jetzt um 750/96 Zoll nach
             unten bewegt!"
 58:  Print
 59:  Gosub Delay3
 60:  '
 61:  '
 62:  ' jetzt 800 Schritte (1/120 inch) nach links als
      Absolutbewegung
 63:  ' Grundcode für horiz. Bewegung nach links in Zeichen
 64:  Zeichen=&HE0
 65:  Schrittzahl=800
 66:  Gosub Dir_bewegung
 67:  Print "Ich habe den Schlitten jetzt um 800/120 Zoll
             nach links bewegt!"
 68:  Print
 69:  Gosub Delay3
 70:  '
 71:  '
 72:  ' jetzt 750 Schritte (1/960 inch) nach oben als
      Absolutbewegung
 73:  ' Grundcode für vert. Bewegung nach oben in Zeichen
 74:  Zeichen=&HF0
 75:  Schrittzahl=750
 76:  Gosub Dir_bewegung
 77:  Print "Ich habe das Papier jetzt um 750/96 Zoll nach
             oben bewegt!"
 78:  Print
 79:  Gosub Delay3
 80:  '
 81:  '
 82:  Print
 83:  Input "Bitte Return-Taste drücken ",A$
 84:  ' Jetzt Schreibteilunng definieren
 85:  Schreibteilung=12
 86:  Gosub Def_teilung
 87:  '
 88:  ' Jetzt 10 Zeichen drucken
 89:  For Schleife=1 To 50
 90:    Tr_code=65
 91:    Print "Ich drucke jetzt ein Zeichen ab !!!!!!!"
 92:    Gosub Drucken
 93:    Gosub Delay2
 94:    Gosub Delay
 95:  Next Schleife
 96:  Schrittzahl=50*120/Schreibteilung
 97:  Gosub Cr_lf
 98:  Print
 99:  Input "Bitte Return-Taste drücken",S$
100:  '
101:  ' Jetzt 10 Zeichen in Fettschrift drucken
102:  Schreibteilung=13.333
103:  Gosub Def_teilung
104:  For Schleife=1 To 50
105:    Tr_code=65
106:    Print "Ich drucke jetzt ein Zeichen in Fettschrift !!"
107:    Gosub Drucken_fett
108:    Gosub Delay2
109:    Gosub Micro_step
110:    Gosub Delay2
111:    Gosub Drucken
112:    Gosub Delay2
113:  Next Schleife
114:  Schrittzahl=50*120/Schreibteilung
115:  Gosub Cr_lf
116:  '
117:  Print
118:  Input "Bitte Return-Taste drücken",Z$
119:  Gosub Offline
120:  Print "OFFLINE"
121:  Restore
122:  Stop
123:  '
124:  '
125:  '
126:  Procedure Les
127:    Aux=Peek(&HFFFA2B)
128:    Erg=Peek(&HFFFA2F)
129:  Return
130:  '
131:  Procedure Schr
132:    Gosub Delay
133:    Do
134:      Stat=Peek(&HFFFA0)
135:      Exit If Stat=0
136:      Print "muß warten !!!!!!!!!!!!!!!!"
137:    Loop
138:    Spoke &HFFFA2F,Zeichen
139:  Return
140:  '
141:  Procedure Delay
142:    For Del=0 To 200
143:    Next Del
144:  Return
145:  '
146:  Procedure Delay2
147:    For Del=0 To 1000
148:    Next Del
149:  Return
150:  '
151:  Procedure Delay3
152:    For Del=0 To 10000
153:    Next Del
154:  Return
155:  '
156:  Procedure Online
157:    Nochmal:
158:    Flag=1
159:    Do
160:      Read Zeichen
161:      Exit If Zeichen=&HFF
162:      Merk=Zeichen
163:      Gosub Schr
164:      Gosub Delay
165:      Gosub Les
166:      Print Hex$(Merk),Hex$(Erg)
167:      If Erg=&HA4
168:        Flag=1
169:      Endif
170:    Loop
171:    Data &a0,0,&a1,0,&a4,0,&a2,0,&ff
172:    Restore
173:  Return
174:  '
175:  Procedure Drucken
176:    ' hier wird ein Zeichen ausgedruckt
177:    Zeichen=Tr_code And &H7F
178:    ' Typenradcode uebertragen
```

```
179:    Gosub Schr
180:    Zeichen=&H17 Or &H80
181:     ' Richtungserkennung und Abschlagstaerke uebertragen
182:    Gosub Schr
183:  Return
184:  '
185:  Procedure Drucken_fett
186:    ' hier wird ein Zeichen ausgedruckt
187:    Zeichen=Tr_code And &H7F
188:    ' Typenradcode uebertragen
189:    Gosub Schr
190:    Zeichen=&HF
191:    ' Bewegungsunterdrueckung und Abschlagstaerke
        uebertragen
192:    Gosub Schr
193:  Return
194:  '
195:  Procedure Dir_bewegung
196:    ' befehlscode steht in variable "Zeichen"
197:    ' Schrittzahl steht in Variable "Schrittzahl"
198:    Msb=Int(Schrittzahl/256)
199:    Lsb=Schrittzahl-Msb*256
200:    Zeichen=Zeichen Or Msb
201:    Gosub Schr
202:    Zeichen=Lsb
203:    Gosub Schr
204:  Return
205:  '
206:  Procedure Def_teilung
207:    Zeichen=&H80
208:    Gosub Schr
209:    Zeichen=Int(120/Schreibteilung)
210:    Gosub Schr
211:  Return
212:  '
213:  Procedure Cr_lf
214:    Zeichen=&HE0
215:    Gosub Dir_bewegung
216:    Schrittzahl=16
217:    Zeichen=&HD0
218:    Gosub Dir_bewegung
219:  Return
220:  '
221:  '
222:  Procedure Offline
223:    Do
224:      Read Zeichen
225:      Exit If Zeichen=0
226:      Merk=Zeichen
227:      Nochmall:
228:      Gosub Schr
229:      Gosub Delay
230:      Zeichen=0
231:      Gosub Schr
232:      Gosub Delay
233:      Gosub Les
234:      For L=0 To 10
235:        Gosub Delay
236:      Next L
237:    Loop
238:    Data &a3,&a0,0
239:  Return
240:  '
241:  Procedure Micro_step
242:    Zeichen=&H83
243:    Gosub Schr
244:    Zeichen=1
245:    Gosub Schr
246:  Return
```

```
Stückliste für Gabriele 9009-Interface

Kondensatoren

2 x 10µF / 15V (Tantal)
2 x  4,7µF / 15V (Tantal)

1 x 8-pol. DIN-Stecker (Schreibmaschine)
1 x 25-pol. Sub-D-Buchse (Serieller Port ATARI ST)

1 x IC MAX 232
```

*ENDE*



# Einzelheft- u. Monatsdisketten Bestellung

ST-Computer können Sie direkt beim HEIM-VERLAG zum Einzelheft-Preis von DM 6,— (1986 + 1987) DM 7,— (1988) nachbestellen. Bearbeitung nur gegen beigefügten Scheck über den entsprechenden Betrag (keine Überweisung).

| Jan. | Febr. | März | April | Mai | Juni | Juli/Aug. | Sept. | Okt. | Nov. | Dez. | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | ■ | ■ | ■ | ■ | ■ | | | | | | 1986 = DM |
| ■ | | | | | | ■ | | | | | 1987 = DM |
| | | | | | | | | | | | 1988 = DM |

+ Gebühr für Porto u. Verp. = DM

(1 Heft DM 2,–, ab 2 Hefte DM 5,–)

☐ Scheck in Höhe zus. DM liegt bei

## Disketten Service

Alle Programme, die in ST-Computer veröffentlicht wurden, sind auf Disketten erhältlich. Die Disketten enthalten die Programme von jeweils 2 ST-Computer-Ausgaben. Bestellen Sie durch ankreuzen die gewünschten Disketten

| Preis je Diskette 28,– DM | Jan./Febr. | März/Apr. | Mai/Juni | Juli/Aug. | Sept./Okt. | Nov./Dez. | Jan./Febr. | März/Apr. | Mai/Juni |
|---|---|---|---|---|---|---|---|---|---|
| | 87 | 87 | 87 | 87 | 87 | 87 | 88 | 88 | 88 |

**Lieferung:** gegen beigefügten Scheck zuzügl. 5,— DM Versandkosten, unabhängig der bestellten Menge.



## Einzelheft- u. Monatsdisketten Bestellung

**Heim Verlag**
Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

# K-OCCAM

## - Der erste Occam-Compiler für den ST

**Im Zusammenhang mit Transputern war immer wieder, auch in dieser Zeitung, von der Sprache Occam die Rede. Seit kurzem gibt es nun einen Occam-Compiler, der sowohl Code für den ST wie auch für eine angeschlossene K-Max-Transputerkarte erzeugen kann.**




Die Firma Inmos, die sowohl den Transputer wie auch die Programmiersprache Occam (nach dem niederländischen Philosophen William von Occam, um 1290-1349, benannt) entwickelt hat, letztere als spezielles Werkzeug zur Programmierung paralleler Probleme, veröffentlichte schon sehr bald nach der ersten Formulierung der Sprachdefinition (1983/84) ein Occam-System, das den Namen Occam-Portakit trägt, weil es sehr leicht auf jeden beliebigen Rechnertyp anzupassen ist. Wie auch das bekannte UCSD-Pascal erzeugt der Compiler einen Pseudocode (der dem Transputer-Assembler zum Verwechseln ähnlich sieht), der dann von einem speziellen Interpreter in der jeweiligen Maschinensprache des Zielrechners ausgeführt werden kann. Dieser Pseudocode ist für einen hypothetischen Rechner, der 'Portakit-Maschine' heißt, gedacht. Seit 1986 gibt es eine erweiterte Occam-Definition, die den Namen Occam 2 trägt. Eine der wichtigsten Erweiterungen ist die Einführung von Fließkommazahlen, die in Occam 1 nicht unterstützt werden. Der Compiler des Portakits ist aber immer noch ein Occam 1-Compiler. Das Kuma K-Occam-System basiert auf dem Compiler des Portakit.

Occam ist eine strukturierte Hochsprache. Variablen sind, wie in Pascal, typgebunden, jede Typenkonvertierung muß bei der Zuweisung deklariert werden. Im Grunde ist die grobe Struktur der von C oder Pascal nicht unähnlich.

In Occam-Programmen gibt es drei Grund-Konstrukte: SEQ, PAR und ALT. Blöcke innerhalb eines SEQ-Konstruktes werden hintereinander ausgeführt, wie in einer ganz normalen Programmiersprache. Solche Blöcke sehen nicht anders aus als in C oder Pascal.

PAR-Konstrukte dagegen deklarieren die darin enthaltenen Blöcke als prinzipiell parallel ausführbar. Zwei innerhalb eines PAR-Konstruktes enthaltene Statements können also parallel ausgeführt werden, wobei es für die Programmierung gleichgültig ist, ob die Blöcke tatsächlich parallel auf verschiedenen Prozessoren oder quasiparallel auf einer Multitasking-Umgebung laufen sollen.

Das ALT-Konstrukt schließlich dient dazu, nach Abfrage einer Bedingung einen von mehreren Prozessen auszuführen, entsprechend etwa einem CASE-Statement in Pascal.

Natürlich ist eine derartige Struktur nur dann sinnvoll, wenn Blöcke innerhalb eines PAR-Statements auch miteinander kommunizieren und gegebenenfalls aufeinander warten können. Dazu gibt es in Occam ein Mitteilungskonzept über sogenannte Kanäle, die z.B. in einer Transputerumgebung durch die Links der Prozessoren repräsentiert werden könnten. Ein Prozeß kann an einen anderen Prozeß eine Nachricht schicken. Dabei wartet der Absender so lange, bis der Empfänger

bereit ist, die Nachricht zu empfangen. Ist der Empfänger früher an einem Punkt angelangt, an dem er eine Nachricht eines anderen Prozesses für die Weiterarbeit benötigt, muß er eben warten, bis der Sender geruht, in Aktion zu treten.
Eine tiefergehende Einführung in Occam würde den Rahmen dieses Artikels sprengen; bitte haben Sie Verständnis dafür, wenn es bei diesen 'Appetithäppchen' bleiben muß.

## Das K-Occam-Paket

K-Occam besteht aus drei Teilen: Dem Occam-Compiler, der von INMOS stammt, dem Pseudocode-Interpreter und einem Assembler, der Pseudocode aus dem Assembler der Portakit-Maschine erzeugt und auch einen einfachen Screen-Editor enthält.
Der Pseudocode-Interpreter ist in drei Versionen zu erhalten: Eine Standard-Version führt Pseudocode-Instruktionen mit 0.1 Millionen Instruktionen pro Sekunde aus. Die zweite Version enthält zusätzliche Bereichsüberprüfungen für den Stack- und Befehlspointer. Dadurch wird die Ausführungsgeschwindigkeit halbiert sowie zusätzliche Betriebssicherheit erreicht, die laut Handbuch besonders für erste Versuche mit langen Programmen sinnvoll ist. Der dritte Interpreter schließlich kann zum Debuggen oder auch nur zum besseren Verständnis dienen: Er zeigt jede Instruktion, die gerade ausgeführt wird, auf dem Bildschirm an.
Der Interpreter ist leicht zu bedienen, es ist ein TTP-Programm, dem man beim Start nur den Namen des zu interpretierenden Programmes (und eventuelle Parameter für dieses Programm) mitgeben muß.
Der Occam-Compiler selbst ist auch nur ein Pseudocode-Programm. Um ihn zu starten, muß man den Interpreter aufrufen und ihm als Parameter den Namen des Compilers (OCCAM) und den Namen des zu compilierenden Programmes (als Sourcecode) sowie optional einige weitere Parameter für den Compiler mitgeben. Als Ergebnis erhält man ein Pseudocode-Programm.
Mit den Optionen kann man den Compiler zum Beispiel dazu bringen, nur einen Syntax-Check durchzuführen, ohne Code zu generieren. Es ist ebenfalls möglich, den Compiler zum Pseudocode als Kommentar auch Assemblerbefehle erzeugen zu lassen, die dann von dem Debugging-Interpreter angezeigt werden. Damit werden dem Benutzer die gerade ausgeführten Befehle etwas übersichtlicher und verständlicher gezeigt als im Pseudocode.
Der Compiler liefert ausführliche Fehlermeldungen. Die Compilierzeiten sind recht flott, allerdings kann der Compiler nicht ohne weiteres lange Programme auf einmal übersetzen; Das liegt daran, daß er immer bestimmte Abschnitte des Programmes auf einmal einliest und übersetzt. Solche Abschnitte sind z.B. Prozeduren oder Variablendefinitionen. Abschnitte dürfen für den Compiler nicht länger als 100 Zeilen sein. Beim Entwurf der Programme muß man darauf achten, am Anfang keine globalen Definitionen zu verwenden, weil alles, was solchen Deklarationen folgt, vom Compiler als ein Abschnitt betrachtet wird, der dann, wie gesagt, nicht länger als 100 Zeilen sein darf. Für manche Anwendung ist das aber keine allzugroße Einschränkung: Der Compiler, in Occam geschrieben, ist im Source-Code ungefähr 15.000 Zeilen lang und kann sich durchaus selbst übersetzen.
Der Interpreter stellt Occam-Funktionen zur Verfügung, die es erlauben, im Source-Level auf Atari-Bildschirm (natürlich nur im Textmodus), Tastatur und Diskettenlaufwerk zubenutzen. Damit ist zumindest für elementare Benutzerkommunikation gesorgt.
Der Assembler versteht den Inmos-Portakit-Assembler (und die dazugehörigen Standard-Mnemonics), der dem Transputer-Assembler sehr ähnlich ist.
Der Screen-Editor erlaubt es, Texte einzugeben. Dabei ist die komplexeste Funktion das Cut&Paste-Feature, aber es ist gerade alles da, was man für elementares Edieren von Programmtexten braucht.
Übliche Assembler-Pseudo-Befehle gibt es natürlich auch, insgesamt ist der Assembler ungefähr so komfortabel wie K-Seka, der Kuma-Assembler für den ATARI ST.
Das Handbuch zu K-Occam hat ungefähr 70 Seiten. Es ist natürlich Englisch und sehr ausführlich, auch eine Einführung in den Portakit-Assembler ist enthalten. Für Occam-Neulinge wird nicht gesorgt; lediglich auf die Occam-Literatur und die Inmos-Originaldokumentation wird hingewiesen. Etwas ungeschickt ist auch, daß das Handbuch aus drei Handbüchern mit jeweils eigener Numerierung besteht, die zu einem Band zusammengebunden sind.
K-Occam ist geradezu ein archaisches Programmpaket. Bedienung und Arbeitsweise erinnern stark an die Computervorzeit, aber das Ganze funktioniert. Meines Erachtens ist dieses Programmpaket auch nicht zum Schreiben vollständiger Anwendungen gedacht sondern zur Einarbeitung in Occam.
Dieser Aufgabe ist es vollständig gewachsen. Man kann vernünftig und schnell mit dem System arbeiten. Wenn man dann die Grundkonzepte von Occam beherscht, ist hoffentlich Occam 2 lieferbar, zusammen mit einem vernünftigen Transputersystem, auf dem sich Occam dann auch so richtig austoben kann.

*CS*

**ENDE**

# Ein neues TOS im Anmarsch!

Über das Betriebssystem des Atari ST ist schon viel geredet, geschrieben und geschimpft worden. Einige ärgerliche Fehler waren immer wieder Stein des Anstoßes heftiger Kritik. Etwa der sogenannte "Unterstrich-Fehler", der bei Eingabe dieses Zeichens ("_") im Kontrollfeld unweigerlich zum Absturz des Rechners führt. In der letzten, jetzt überall ausgelieferten Version des TOS wurden kleinere Haken und Ösen korrigiert. Aber viel mehr auch nicht. Besondere Anwenderwünsche müssen nach wie vor über spezielle Programme realisiert werden. Zum Beispiel das Starten von GEM-Anwendungen direkt nach dem Hochfahren. Seit einiger Zeit ist nun in den USA ein Tool erhältlich, das ein gänzlich neues Desktop mit vielen Zusatzfunktionen auf dem ST darstellt. Es heißt NeoDesk und leistet mehr als alle anderen Kommandointerpreter.

Ob NeoDesk als Stachel im Fleisch der amerikanischen Atari-Zentrale wirkte, oder ob die neue Firmenstruktur ihren Teil dazu beigetragen hat: Jack Tramiel und sein Team haben sich entschlossen, dem ST ein neues TOS zu verpassen, das die eh schon bequeme Benutzeroberfläche noch attraktiver machen soll. Das neue Betriebssystem wird derzeit als Beta-Version an die Atari-Niederlassungen in aller Welt ausgeliefert. Was es damit auf sich hat, berichtete Roy J. Good in den amerikanischen Computernetzen:

1. Das Verschieben von einzelnen Dateien auf der Desktop-Ebene läuft anders und, wie zu hören ist, besser. Man kann jetzt mit nur einem Mauskommando eine Datei bewegen, d.h. in einen Ordner kopieren und gleichzeitig am Ursprungsort löschen, so wie beim Apple MacIntosh. Die Undo-Taste bewirkt den Abbruch von Kopieraktionen. Das Kopieren geht auch mit nur einem Diskettenlaufwerk erheblich schneller, weil nämlich jetzt der gesamte Arbeitsspeicher dazu genutzt wird. Endlich! Sollte am Zielort eine gleichnamige Datei schon vorhanden sein, erscheint eine Auswahlbox mit "Kopieren/Überspringen/Abbruch". Den Namen eines Ordners verändert man im neuen TOS mit der Funktion "Zeige Info".
2. Neugestaltet wurde die Objektauswahlbox. Nach dem Aufruf finden Sie nun 16 Buttons für unterschiedliche Laufwerke. Die Box "merkt" sich, an welcher Stelle der Dateienliste sie bei der letzten Anwahl stand. Ein laufendes Programm kann eine Titelzeile an das Auswahlfenster schicken, so daß deutlich ist, welche Operation gerade abläuft.
3. Alle Schwierigkeiten, die sich aus zu vielen Dateien und Ordnern auf einer Festplatte ergeben (40 Ordner-Problem), sollen mit dem neuen TOS endgültig behoben sein. Dazu wurde die Begrenzung auf 100 Dateien pro Wurzelverzeichnis aufgehoben und die Malloc-Funktion verändert.
4. Beschleunigte Disketten- und Festplattenzugriffe garantiert nach der TOS-Beschreibung ein spezieller FAT-Lesemodus. Die FAT ("File Allocation Table") verwaltet das Inhaltsverzeichnis der Diskette, und die bisherige Zugriffsart gilt als recht langsam. Turbodos greift zum Beispiel ebenfalls in die FAT ein und erzielt damit ein höheres Schreib- und Lesetempo. Der interne Buffer für Sektoren wurde vergrößert und kann mit einem zusätzlichen Cache-Programm beliebig hochgesetzt werden.
5. Kleinere Verbesserungen gibt es bei einigen AES-Aufrufen. Dies ist für Programmierer relevant. Alle Anwender wird interessieren, daß das neue TOS den Autostart von GEM-Programmen gestattet und einen Software-Reset durch gleichzeitiges Drücken von CTRL/ALT/DELETE erlaubt.
6. Neben dem schon erwähnten Cache-Programm gehört zum Lieferumfang des neuen TOS ein ebenfalls modifizierter Festplattentreiber, der Kapazitäten bis zu 60 MByte verwaltet.

Wie geht es nun weiter mit dem vorläufigen, neuen TOS? Atari-USA hat recht rigide Richtlinien zum Testen dieser Beta-Version erlassen. Jede Vertretung dürfe einige Exemplare unter strengen Bedingungen weitergeben. Adressaten sind wohl in erster Linie Journalisten und Firmen. Wer ein Testmuster erhält, muß sich schriftlich verpflichten, einen wöchentlichen Report (jeden Freitag!!) an Atari zu schicken und ansonsten Stillschweigen zu wahren. Das hört sich etwas kleinkariert und verschroben an, macht aber dennoch Sinn. Nichts wäre dem ST abträglicher, als wenn die Beta-Version heilloses Chaos in Zeitschriften, Klubs und Mailboxen herbeiführen würde. Aber ich denke, daß möglichst viele ST-Anwender ihre Erfahrungen und Wünsche in das neue TOS einbringen sollten. Die Atari-Mailbox in Raunheim (Tel.: 06142/21161) wäre etwa ein geeigneter Ort.

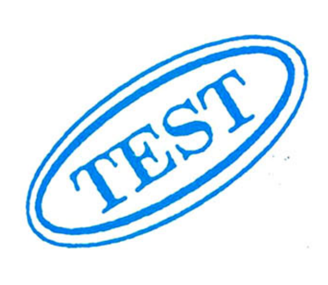
Nicht zuletzt scheint mir das eigentlich Wichtige am neuen TOS zu sein, daß die Atari-Entwickler endlich aus ihrem selbstgeschaffenen Elfenbeinturm heraustreten und ein breiteres Publikum zum Mitwirken auffordern. Auf diese Art und Weise kann es möglich sein, das Betriebssystem vorab in vielfältigen Konfigurationen zu testen, Fehler unwahrscheinlicher zu machen und damit etwas Ausgereiftes auf den Markt zu bringen. Das war gerade bei Atari nicht immer so. Aber seitdem Charles Babbitt die Oberaufsicht über die neue Tochter "Atari-Computers" erhalten hat, sind wohl Umstrukturierungen im Gange, die gegenüber der bisherigen Firmenpolitik nur als positiv bewertet werden können.

Noch einmal: wichtig ist jetzt, daß Fehler im TOS aufgezählt und Atari mitgeteilt werden. Wünschenswert wäre, daß Vorschläge, Ideen und Anregungen via Raunheim auf den Schreibtisch von Joe Ferrari gelangen. Es liegt jetzt an uns, den Atari ST ein gutes Stück besser zu machen.

*Michael Spehr*

**Da freut sich der Programmierer. Neue und veränderte Funktionsaufrufe im TOS:**

- "wind_update(FALSE)" wird gesetzt, wenn eine Anwendung nicht ordnungsgemäß verlassen wurde;
- nach nicht-edierbaren Texten in Dialogboxen können Eingabefelder folgen;
- bei der "vq_extent" wurde der Pixelfehler bei einigen 270 Grad Rotationen behoben;
- "frename" erlaubt das Umbenennen eines Ordners;
- das "Archive bit" wird voll unterstützt;
- neue "media-change"-Erkennung;
- Routine für "zeige/drucke Datei" wurde völlig umgeschrieben;
- ebenso die Funktionen "Ccon*" und "Malloc";
- Diskettenformat wurde MS-DOS weiter angeglichen;
- die Objektauswahlbox handhabt auch lange Verzeichnisketten korrekt;
- Soft-Reset und Kaltstart über Tastatur;




# ST-KONTOR

## Programme für kaufmännische Anwendungen

**Der Sybex Verlag hat eine Reihe von fünf Programmen für den kaufmännischen Bereich für den ST entwickelt und unter der Titelbezeichnung ST Kontor herausgebracht. Als erstes Programm existiert ein TOS-Manager, der den GEM-Desktop ersetzen und seine Funktionen für den Programmanwender vereinfachen soll. Hauptsächlich besteht die Funktion des TOS-Managers darin, die vier Hauptprogramme des ST-Kontor-Pakets zu bedienen.**




In diesem Bericht werden der TOS-Manager und die ST-Kontor-Kundenverwaltung vorgestellt. Die Finanzbuchhaltung wird gerade in einer neuen Version fertiggestellt, und ein Test erfolgt in der nächsten ST-Computer zusammen mit dem Modul ST-Kontor-Lager und Fakturierung.

## Der TOS-Manager

Bei umfangreichen Programmpaketen ist es sinnvoll, alle Programmteile von einem Hauptprogramm aus zu starten oder zusätzliche Hilfsprogramme aus diesem Programm aufzurufen. Bei der ST-Konto-Reihe soll dieses Ziel durch den TOS-Manager erledigt werden, der die einzelnen Module aus der ST-Kontor-Reihe startet.

Nach dem Einlegen der TOS-Manager-Diskette erscheinen die Programme TOSMANAG.PRG, FILTER.PRG und VDISK.PRG sowie einige Textdaten.

Die Datei NOTIZ.TXT enthält normalerweise noch Anmerkungen zum Programm, ist aber in der vorliegenden Version leer (wer seinen Rechner mal endlos laufen lassen will, darf die Datei jetzt anklicken).

Nach dem Start des TOS-Managers erwartet das Programm die Eingabe von Systemzeit und Datum. Dabei wird die aktuell im System gespeicherte Zeit als Vorauswahl angezeigt.

Es erscheint die Menüzeile, und auf dem Schreibtischhintergrund werden die Laufwerke A: bis J: dargestellt, wobei die tatsächlich angeschlossenen selektiert werden können. Durch Anklicken eines Laufwerkes wird ein Fenster mit den dazugehörenden Dateinamen geöffnet. Durch Anklicken eines Dateinamens kann man die vom Desktop her bekannten Funktionen 'Anzeigen', 'Kopieren', 'Umbenennen', 'Drucken', 'Ausführen' und 'Löschen' erreichen. Allerdings gestalten sich die Operationen nicht so elegant, wie vom DESKTOP her gewohnt, z.B. muß zum Kopieren das Laufwerk, auf das eine Datei kopiert werden soll, über die Dateiauswahlbox angewählt werden.

Zusätzlich existiert im 'Ordner Menü' die Möglichkeit, Ordner anzulegen, zu kopieren und zu löschen. Eine weitere Funktion wird zwar offiziell nicht vom Atari-Betriebssystem unterstützt, aber sie ist manchmal sehr nützlich. Unter dem Eintrag 'Diskette' können Disketten kopiert und formatiert werden. Es ist möglich, die Disketten so zu formatieren, daß sie 50 K-Byte mehr Speicherplatz zur Verfügung haben. Allerdings sollte man diese Möglichkeit nur anwenden, wenn das eigene Laufwerk auch mitspielt (einige Laufwerke mögen diese Übersättigung nämlich nicht so gern).

Bei der Anzeige eines 'Diskinfos' kann zusätzlich zu der Funktion vom Desktop der Name der Diskette geändert werden. Recht nützlich ist die Funktion der Laufwerke A: und B:, Dateien zu restaurieren, die versehentlich gelöscht worden sind. Das gelingt auf dem Atari allerdings nur direkt nachdem eine Datei gelöscht wor-

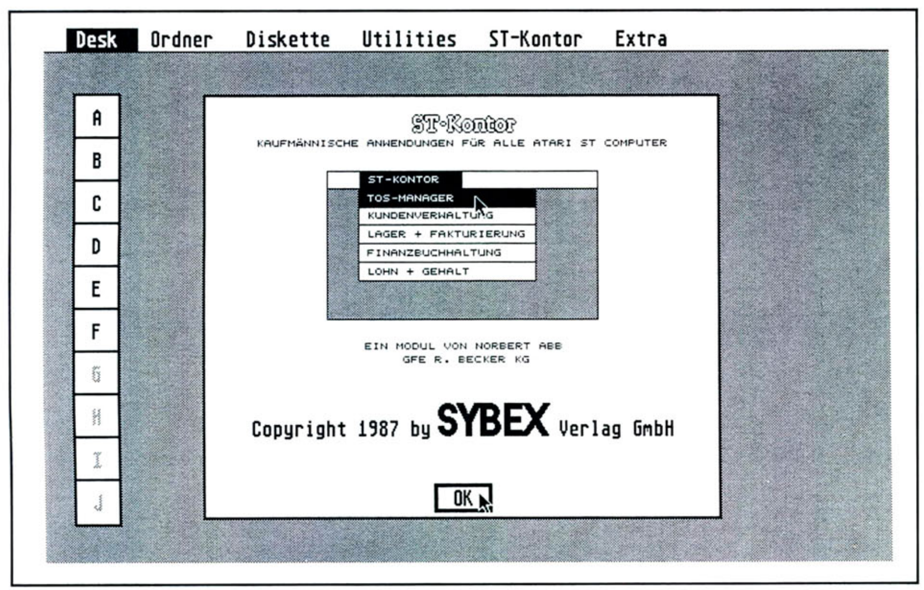



*Bild 2: Laufwerksleiste und Copyrightmeldung*

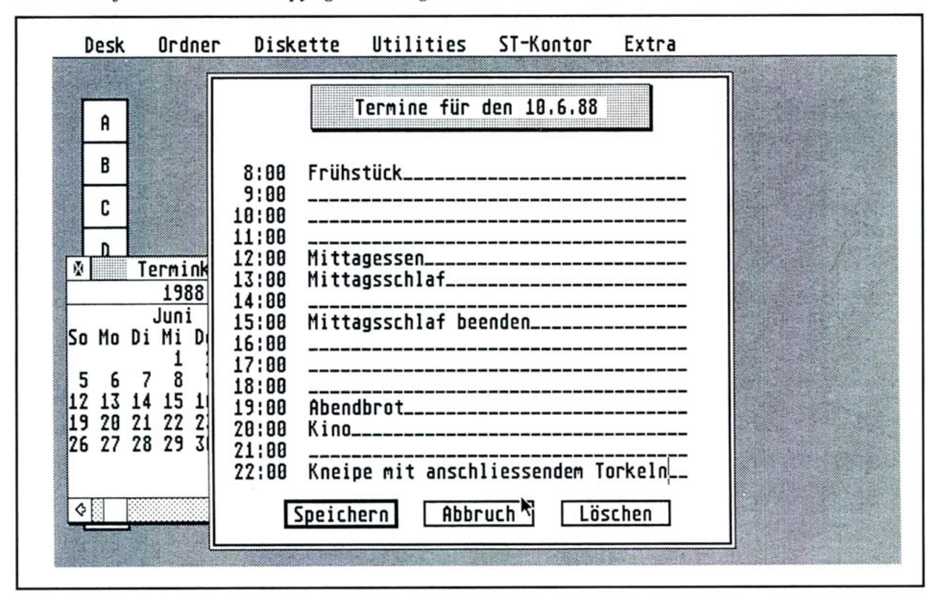



*Bild 3: Terminkalender*

den ist. Als nächster Menüpunkt erscheinen die Utilities, also kleine Hilfsprogramme, die der TOS-Manager zur Verfügung stellt. Als erstes läßt sich eine einfache RAM-Disk (VDISK.PRG) installieren, die auch nach Verlassen des TOS-Managers noch aktiv ist. Ferner ist ein kleiner Notizblock verfügbar, der es ermöglicht, in einer Dialogbox 10 Zeilen mit jeweils 50 Buchstaben zu notieren. Diese Notiz steht danach auch den anderen ST-Kontor-Programmen zur Verfügung.

Recht pfiffig ist ein eingebauter Terminkalender, bei dem zu jedem Tag zwischen den Jahren 1980 und 2000 eine Termin zwischen 8.00 und 22.00 Uhr zu jeder vollen Stunde eingetragen werden kann. Beim Start des TOS-Managers werden die Termine abgefragt und angezeigt. Weniger begeistern konnte die ASCII-Tabelle (die zusätzlich auch im Handbuch ist), deren Darstellung auch für Anwender dieses Programmes wahrscheinlich nicht sehr bedeutend ist. Als letztes Tool ist ein 10-stelliger Taschenrechner (mit Klammern, ohne algebraische Logik) eingebaut, der über eine Kommandozeile Rechenausdrücke erhalten und berechnen kann. Falls der Terminkalender oder die ASCII-Tabelle mal nicht erscheinen, braucht übrigens keine Unruhe aufzukommen, es ist lediglich kein Fenster mehr frei (bzw. vier Fenster sind vom Programm belegt).

Der nächste Menüpunkt 'ST-Kontor' soll es ermöglichen, die Programme der ST-Kontor-Reihe per Tastendruck zu starten. Dafür wird einmalig deren Ort auf den Laufwerken festgelegt (in der Pfadliste), und danach startet der TOS-Manager automatisch das gewählte Programm. Er verfügt außerdem über einen 'Command-Line-Interpreter' - sprich CLI, mit dem Befehle über die Tastatur, wie auf Maschinen mit dem Betriebssystem MS-DOS, ausgeführt werden können. Arbeiten kann man damit schon, zeitgemäß ist es im kommerziellen Softwarebereich sicherlich nicht. Auch die Autoren haben sich auf eine Nennung und Kurzbeschreibung der knapp 100 Befehle in der Anleitung beschränkt.

Neben den Funktionen, die per Menü auszuwählen sind, existieren noch ein Funktionstastenmenü, bei dem zehn weitere Funktionen zur Verfügung stehen. Zum einen können hier die Pfade (der Platz, an dem sich ein bestimmtes Programm befindet) eingestellt, zum anderen eine Textverarbeitung gestartet oder bestimmte Systemparameter gesetzt werden.

Etwas weniger sinnvoll ist der Virus-Filter (FILTER.PRG: ein Programmteil, der Programme heraussuchen soll, die vom Virus-Constructions-Programm infiziert sind), der prinzipiell nur auf einen Virus anspricht (hat auch beim Claus nicht geholfen. Gute Besserung übrigens.) und sogar bei der nicht infizierten Originaldiskette von Sybex nicht richtig funktioniert hat (Bus Error at line 123 in MAIN PROGRAMM). Bei der Funktion Pfadangaben ist es nicht sehr schön, daß sich das Programm bei einer schreibgeschützten Diskette total verabschiedet und sich ohne Druck auf den RESET-Knopf nicht mehr wiedermeldet.

Von der Idee her ist der TOS-Manager sicherlich zu begrüßen, aber die Realisierung ist noch zu verbessern, insbesondere in Bezug auf bessere und damit einfachere Bedienerführung sowie in punkto Betriebssicherheit des ganzen Programms.

## ST-Kontor-Kundenverwaltung

Die ST-Kontor-Kundenverwaltung ist das erste Modul in dieser kaufmännischen Serie, das, wie der Name schon sagt, für die Verwaltung von Kunden- und anderen Adressen zuständig ist. Alle von diesem Programm verwalteten Adressen können von den anderen Modulen - Finanzbuchhaltung, Lohn und Gehalt - aufgerufen werden. Das Programm läßt sich direkt von Harddisk bzw. Diskette oder vom TOS-Manager starten, wobei bei größeren Datenbeständen oder bei der Benutzung von mehreren Modulen eine Harddisk empfehlenswert ist.

Beim Start der ST-Kontor-Kundenverwaltung erscheint nach kurzer Wartezeit eine Eingabebox, in der man Datum und

Uhrzeit ändern bzw. bestätigen kann.

Die ersten Menüpunkte behandeln das Eingeben und bearbeiten von Adressen. Unter der Eingabe wird hier verstanden, die Adressen und zugehörige Daten einzugeben und abzuspeichern. Es stehen je 5 Datenmasken zur Verfügung, wobei jedoch nur in der ersten Maske, der Kundenstammdatenmaske, mindestens ein Feld ausgefüllt werden muß. Einträge in den restlichen Datenmasken können bei Bedarf erfolgen. Etwas gewöhnungsbedürftig ist die Bedienung mit Maus und Tastatur bzw. den Tastenkombinationen, sowie das Blättern zwischen den einzelnen Datenmasken. Im Handbuch findet man allerdings eine Übersicht über Maus- und Tastaturbefehle, die man anfänglich zum Orientieren braucht.

In der Stammdatenmaske wird für jede Adresse eine Adreßart bestimmt, die man durch Mausbedienung auswählt und bei der zwischen acht Kriterien gewählt werden kann: Personal, Lieferant, Kunde, Interessent, Privat, Versicherung, Behörde und Sonstiges. Falls mal vergessen wird, die Adreßart anzugeben, wählt das Programm automatisch die Adreßart Sonstiges aus.

Die zweite Maske, die Kundenzusatzmaske, enthält Felder für zwei Bankverbindungen, zwei Zeilen für Bemerkungen sowie Liefer- und Rechnungsadresse.

Die dritte Maske enthält ein Notizfeld, in dem zwölf Zeilen für Texte zur Verfügung stehen, die auch in Listen oder Serienbriefen berücksichtigt werden können.

Die vierte Maske dient zur Datenselektion. Es besteht hier die Möglichkeit, bis zu zwanzig Selektionskriterien für eine Adresse einzutragen, um z.B. einen bestimmten Kundenkreis über eine Funktion zu selektieren.

Die fünfte, die Datenzusatzmaske, ist von der Adreßart abhängig.

Für Interessenten, Kunden und Lieferanten können Zahlungsbedingungen, die hauptsächlich für die Fakturierung interessant sind, eingegeben werden. Weiterhin besteht die Möglichkeit, Telex, Teletex, Telefax, BTX und Mailboxnummern zu speichern, um für solche Einrichtungen Serienschreiben zu verschicken.

Die Datenmaske für Personal und Mitarbeiter beschränkt sich auf Angaben, die für Sozialversicherung und andere Beiträge notwendig sind.

Die angelegten Adressen kann man sich auf verschiedene Arten ausgeben lassen. Man muß sich dazu mit einer Tastenkombination eine Auswahlbox holen, in der die Möglichkeiten zum Druck bzw. zur Ausgabe dargestellt sind. Die Ausgabe erfolgt auf Drucker oder Datei, wobei Liste, Brief oder Etikette Ausgabearten darstellt. Zusätzlich hat man die Wahl zwischen dem Druck aller Adressen oder einer Adreßauswahl. Dabei muß das richtige Dokument durch Pfadangabe bestimmt werden. Die Formulare sind nur für 1st_Word bzw. 1st_Word plus angepaßt.

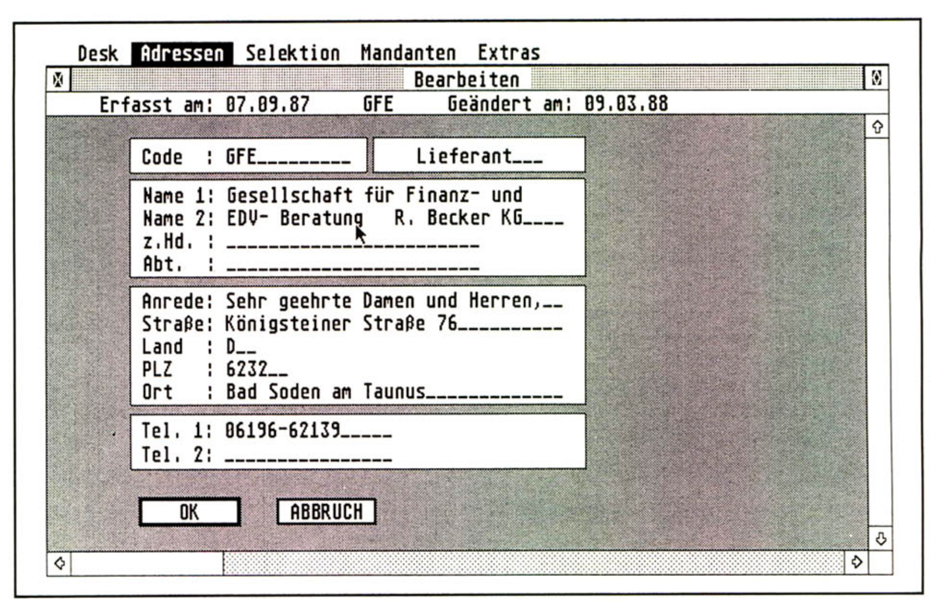



*Bild 4: Hauptmaske der Kundenverwaltung*

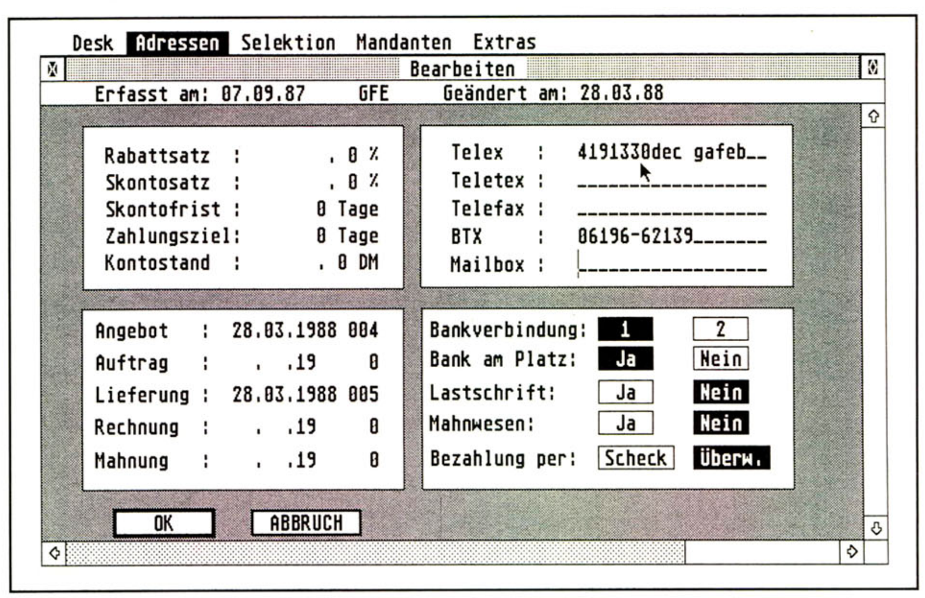



*Bild 5: Fakturamaske der Kundenverwaltung*

Selektionen erstellen und bearbeiten ist ein weiterer Menüpunkt der ST-Kontor-Kundenverwaltung; sie selektiert Adreßgruppen. Es stehen bis zu 999 Selektionskriterien zur Verfügung, bis zu 20 Stück für jede Adresse (in der Selektionsmaske). Bei der Selektierung werden alle vorhandenen Adressen auf die angewählten Kriterien hin überprüft. Die Selektionen lassen sich jeweils noch durch logische Operationen miteinander verknüpfen. Da diese Anwendung mandantenfähig ist, dürfen die Funktionem 'Mandant anlegen' und 'Mandant wechseln' nicht fehlen. Bei Anwählen der ersten Funktion erscheint eine Dateiauswahlbox, in der angegeben wird, wo sich der neue Mandantenordner befindet. Der ST-Kontor-Kundenverwaltung legt einen Ordner unter dem angegebenen Namen an und installiert alle notwendigen Dateien hinein. Wenn der Mandant gewechselt wird, muß in der erscheinenden Dateiauswahlbox nur der entsprechende Ordner mit der Mandantenbezeichnung ausgewählt werden. Leider ist bei der Bearbeitung eines Mandanten nicht sichtbar, für welchen gerade gearbeitet wird.

Die mitgelieferten Extras, Funktionstastenbelegung und Notizblock entsprechen dem TOS-Manager. Zusätzlich hat man die Möglichkeit, sich Hilftexte einblenden zu lassen. Diese Funktion wird auch über Extra aufgerufen. Einmal akti-

*Bild 6: Wahl der Selektionskriterien bei der Kundenverwaltung*

viert, wird beim Anwählen von Menüpunkten erklärt, was zu tun ist. Dieses Hilfstexte können für den eigenen Gebrauch geändert werden.
In den mitgelieferten Utilities befindet sich ein Paßworteditor, der für die Belegung einzelner Arbeitsbereiche durch Paßworte zuständig ist. Große Dateien kann man in zwei Mandaten unterteilen, getrennt bearbeiten und wieder zusammenfügen. Außerdem wird ein Backup-Programm mitgeliefert, bei dem entweder der gesamte Datenbestand oder nur geänderte Dateien gesichert werden können.
Das Programmpaket ST-Konto-Kundenverwaltung beinhaltet für einen Preis von DM 149,- eine Programmdiskette und ein Handbuch, das aus etwas mehr als 100 Seiten besteht. Die Beschreibung der einzelnen Programmpunkte und Funktionen ist recht knapp geraten. Sie enthält eine kurze Erklärung der Funktionen, einige Anwendungsbeispiele sowie eine Übersicht der Maus- und Tastaturbefehle. Im Anhang werden die Dateiformate und die Platzhalter für die Formulare dargestellt.
Aus der ST-Kontor-Reihe sind bis heute der TOS-Manager, die ST-Konto-Kundenverwaltung, die ST-Kontor-Finanzbuchhaltung und das Modul ST-Kontor-Lager und Fakturierung auf dem Markt. Da wir Ihnen keines dieser Module vorenthalten wollen, es aber inzwischen neue Versionem gibt, möchten wir den Test für heute abschließen. In der nächsten Ausgabe können Sie dann einen ausführlichen Test über die Module ST-Kontor-Finanzbuchhaltung, ST-Kontor-Lager und Fakturierung lesen.

*Karen Steger / OJO*

**ENDE**

# QUICKTIPS ZU 1ST_WORD

**Diesen Artikel (eigentlich ist es mehr eine Sammlung von Tips und Tricks) wollen wir der wohl am weitverbreitetsten Textverarbeitung auf dem ATARI ST, 1st_Word, widmen. Im Laufe der Zeit hat sich bei uns in der Redaktion eine große Menge Tips zu allen möglichen Anwenderprogrammen angesammelt, die wir Ihnen natürlich nicht vorenthalten möchten. Dabei gehen wir in einer Art Frage-und-Anwort-Spiel vor, da dies am übersichtlichsten ist. Wir wollen uns für die vielen Zuschriften bedanken und gleichzeitig auch wieder Sie, lieber Leser, zum Senden solcher Quicktips aufrufen.**

## Wie bekommt man einen Textausschnitt von einem Textfenster in ein anderes?

Textausschnitt als Block markieren, ausschneiden. Das andere Fenster aktivieren und den Cursor an der Einfügestelle postieren den Block einfügen. Besonders wichtig ist diese Lösung, wenn man Fußnotentexte wieder in den ursprünglichen Text einfügen möchte. Der Fußnotentext wird bei Abspeicherung als ASCII-Text nicht(!) umgewandelt.

## Diskette ist voll. Wie den Text abspeichern, ohne großartig Dateien zu löschen?

Diese Frage ist wichtig, wenn man einen Text bearbeitet (nicht neu angelegt!), hat, und es fehlen gerade ein paar Byte. Man löscht einfach den bearbeiteten Text auf der Diskette incl. dem BAK-File. Speichert man nun den Text ab, findet sich in der Regel noch Platz. Das platzverschwendende BAK-File wird nicht angelegt, da der Text als neu geschriebener behandelt wird.

## Wie erreicht man alle Fußnotentexte ohne große Sucherei?

Textcursor an das Textende stellen und den Buchstaben "e" nach unten suchen. "e" taucht sehr häufig auf, dadurch ist die Wahrscheinlichkeit groß, alle Fußnoten zu erreichen.

## Wie kann man Sonderzeichen in den Fußnotentext einfügen?

Normalerweise einfach, aber das Fußnotenfenster läßt die Bearbeitung anderer Fenster nicht zu. Also: Textfenster verkleinern, so daß das Sonderzeichenfeld sichtbar wird. Fußnote aufrufen und Fußnotenfenster verschieben. Nun sollte das Sonderzeichenfeld freiliegen und die Sonderzeichen können eingefügt werden. Eine weitere Möglichkeit wäre das Programm EXTKEY aus der Programmierpraxis der ST Computer 6/88.

## Wohin mit Texten zur Zwischenbearbeitung?

Neues Dokument anfangen, etwa CLIP.DOC. Dorthin kann man alle Texte zur Zwischenbearbeitung bringen. Anschließend wird das Fenster CLIP.DOC ohne Abspeicherung geschlossen. So hat man zeitweilig ein CLIP-Board, ohne Diskettenplatz zu belegen. Will man allerdings nur einen Text zwischenspeichern, erledigt man dies am besten mit der Ausschneide- und Einfügefunktion.

## Fußnote ohne Leerzeile zur Abtrennung

Auch das ist möglich, trotz des Fehlers bei Neuformatierung. Durch das Drücken der Taste F10 wird automatisch eine Leerzeile an die Fußnote angehängt. Drückt man anschließend BACKSPACE, wird diese Leerzeile wieder gelöscht.

## Ungewolltes Zusammenfügen zweier Absätze bei "Neu Format"

Dieser Fehler tritt dann auf, wenn hinter dem letzten sichtbaren Zeichen (meistens ein Punkt) noch ein oder mehrere Leerzeichen eingegeben worden sind. Eine einfache Zeile endet bei 1st_Word mit einem Leerzeichen (ASCII 30), gefolgt von der Zeilenendemarkierung (ASCII 13, 10). Die letzte Zeile eines Absatzes zeichnet sich dadurch aus, daß vor der Zeilenendemarkierung **kein** Leerzeichen steht. Wenn man also ein Leerzeichen eingibt, bevor man die Return-Taste drückt, wird das Absatzende ignoriert. Abhilfe schafft: Einfach mit dem Cursor unter den Absatz gehen und einmal Cursor nach links drücken. Schon kann man sehen, ob noch ein unsichtbares Leerzeichen vorhanden ist. Wenn ja, dann muß man es einfach löschen.

## Lästiges Ausschalten der Attributfunktionen über Menüleiste

Hat man zum Beispiel Superscript eingeschaltet und will später normal weiterschreiben, so kann man sich den lästigen Weg über die Menüleiste sparen, um diese Attributfunktion wieder auszuschalten. Ein einfacher Druck auf die Escape-, Help oder Undotaste genügt (nur eine der Tasten).

## Steuerbefehle für den Drucker

Da Farbbänder recht teuer sind, versucht man, sie möglichst lange zu benutzen. Wenn man die Schrift normalerweise kaum noch lesen kann, hilft es eine Zeit lang, die Texte mit der Druckeroption "Doppeldruck" auszudrucken. Leider

fehlt 1st_Word diese Funktion. Es besteh die Möglichkeit, aus der Zeichensatztabelle ein sonst nie genutztes Zeichen auszusuchen und im Druckertreiber (*.HEX) anstelle des Zeichencodes den HEX-Code für Doppeldruck (siehe Druckerhandbuch) zu schreiben. Setzt man nun dieses Zeichen als allererstes in einen zu schreibenden Text, wird es als "Escape (bzw.Control) -Code interpretiert und daher nicht gedruckt. Anstelle dessen wird die im Druckertreiber für dieses Zeichen eingetragene Druckfunktion ausgelöst. Selbstverständlich kann man so auch alle anderen Druckeroptionen von 1stWord aus nutzen, z.B. doppelte Höhe von Zeichen, anderer Zeilenabstand (1,5), invertieren, etc.!

## In langen Texten scrollen

Bei sehr langen Texten empfiehlt es sich, die Kapitel auf verschiedene Dateien zu verteilen. Man kann ja bis zu vier Windows gleichzeitig bearbeiten. So kann man zwischen den Fenstern sehr viel schneller hin- und herschalten, als wenn man in einer zu langen Datei scrollen müßte.

## Wörter aus dem Wörterbuch entfernen

Man startet 1st_Word und läd das Wörterbuch. Dann ruft man die Option "Wörterbuch bearbeiten" auf. Ein Wort, das falsch geschrieben oder völlig fehl am Platz ist, wird angeklickt. Dadurch erscheint es in der oberen Textzeile. Mit der Pfeiltaste links wird der Cursor an den Wortanfang geschoben. Nun wird ein "y" eingegeben. Aus dem Wort "sexbombe" wird "ysexbombe". Dieses Wort wird nun mit der Option "Wort hinzufügen" in das Zusatzwörterbuch aufgenommen.
Anschließend wird das "y" wieder entfernt,. Mittels RETURN-Taste ruft man"Suchen" auf. Dadurch befindet man sich wieder auf jener Wörterbuch-Seite, die man zuletzt bearbeitet hatte. Nun kommt, nach einem evtl. Weiterblättern, das nächste dumme Wort an die Reihe.
Alle "y"-Worte stehen schließlich in einem Zusatzwörterbuch, das man am Ende abspeichert. Gerechterweise bekommt es den Namen MIST.SUP. Nun wird MIST.SUP als Datei in 1st_Word geladen. Der nächste Schritt ist dann, daß mit der Option "Ersetzen" alle "y" durch ein "-" ersetzt werden. Danach stehen in MIST.SUP alle überflüssigen Wörter mit einem "-"- Zeichen.
Nachdem MIST.SUP abgespeichert ist, wird mit dem Programm DICMERGE.APP das Hauptwörterbuch SPELLING.DIC mit dem Zusatzwörterbuch MIST.SUP vereinigt. Dabei werden alle Wörter, die in MIST.SUP stehen, aus dem Hauptwörterbuch entfernt.
Zugegeben: Das ist eine langwierige Arbeit. Aber die Prozedur funktioniert. Aufpassen muß man dabei auf solche Wörter, die schon ein "y" enthalten. Hier muß man nach dem automatischen Wandeln von "y" in "-" diese Umwandlung im Wort mit Hand rückgängig machen. Der Aufwand lohnt sich.

## Vorgewählte Formate

Will man sich mehrere Dokumentformate vordefinieren, gibt es eine einfache Möglichkeit. Man legt auf der gleichen Directory-Ebene, auf der sich auch 1st_Word befindet, einen Ordner namens FORMATS an. In diesen kann man nun vordefinierte, leere Dokumente abspeichern. Sie müssen alle den Namen FORMAT mit einer beliebigen Extension (.DOC, .TXT, .BRF usw.) tragen. Öffnet man dann ein neues Dokument, das die gleiche Extension trägt, werden automatisch das Seitenformat und die Linealeinstellung aus der entsprechenden Format-Datei übernommen.

## Probleme beim Abspeichern

Wenn man

- während der Arbeit mit 1st_Word die Diskette wechselt
- den Text auf eine andere Diskette speichern will
- den Text von einer anderen Diskette lädt und ihn anschließend auf der 1st_Word-Diskette speichern will

kann es zu folgenden Problemen kommen:

Es erscheint die Fehlermeldung:

Schreibfehler/Lesefehler Ursache falsche Diskette oder Diskette schreibgeschützt!

Auch wenn man nach dieser Meldung die Original-1st_Word-Diskette einlegt (und gegebenenfalls den Schreibschutz entfernt), ist es immer noch nicht möglich, einen Text zu speichern.

Abhilfe kann folgendermaßen geschaffen werden:

1. Diskette, von welcher der Text geladen wurde (Schreibschutz entfernen!), in das Laufwerk legen.
2. Auf "Speichern als..." klicken.
3. Auf das "Ordnerschließfeld" links oben klicken.
4. Auf "Abbruch" klicken.
5. Text mit "Speichern" abspeichern.

## Probleme beim Ausdruck großer Graphiken

Beim Ausdruck von größeren (etwa Bildschirmgröße) Graphiken mit 1st_Word Plus kann es passieren, daß der Drucker zwischen jeder Graphikzeile eine Leerzeile druckt. Ursache dafür ist das Zeilenlineal unter der Menüleiste. Abhilfe kann geschaffen werden, indem man den Randsteller (eckige Klammer) weiter nach rechts schiebt.

## Ausblenden von Zeichen aus dem Sonderzeichenfeld

Wer stolzer Besitzer eines Panasonic KX-P1081-Druckers ist, wird vergeblich nach einem geeignetem Druckertreiber in der gelieferten Software des Textverarbeitungsprogramms suchen. Der Druckertreiber muß also selbst erstellt werden, was sich als unproblematisch erweist, wenn die mitgelieferte Bedienungsanleitung ausreichend studiert wird und man viel probiert. Nach dem Aufruf des INSTALL.PRG erhält man dann einen entsprechenden Treiber mit der Bezeichnung .CFG.
Wird dieser erstellte Druckertreiber von 1st_Word aus aufgerufen, hat er den Nachteil, daß auf der Zeichensatztabelle des Programmes der gesamte Zeichensatz des ATARI abgebildet wird. Nun sind aber nicht alle Zeichen des ATARI von jedem Drucker auch tatsächlich darstellbar. Bei der Erstellung des Druckertreibers hat man sich damit geholfen, das Zeichen mit einem Nullwert zu versehen, damit es ignoriert wird. Mir scheint es durchaus verwirrend, wenn auf dem Bildschirm Zeichen dargestellt werden, diese auch im Text genutzt werden können, aber nicht auf dem Drucker erscheinen. Was liegt also näher, als diese Zeichen auch auf dem Bildschirm auszublenden? Bei den mitgelieferten Druckertreibern ist dies bereits geschehen. Die Erstellung eigener Druckertreiber läßt diese Möglichkeit jedoch nicht zu. So muß der bereits erstellte Treiber .CFG nachträglich mit einem Disk-Monitor geändert werden.
Bei der Betrachtung des Files des Druckertreibers habe ich festgestellt, daß ein auf dem Bildschirm dargestelltes Zeichen durch 3 Bytes im Treiber festgelegt ist:

geht weiter...

1. Wert - 03
2. Wert - Hex-Wert des dargestellten Zeichens
3. Wert - 00, wenn das Zeichen ignoriert werden soll bzw. Hex-Wert des dargestellten Druckerzeichens

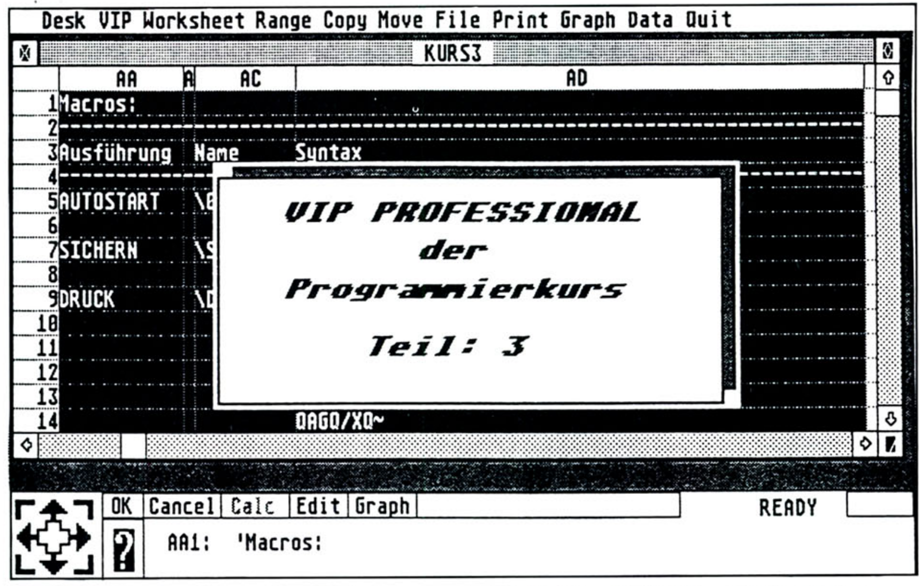
Will man das entsprechende Zeichen ausblenden, müssen die 3 Bytes auf 2 reduziert werden. Dabei muß der 1. Wert dann unbedingt 02 betragen und der zweite den Hex-Wert des auszublendenen Zeichens.
Mit dieser Methode läßt sich die Zeichentabelle von 1st_Word genau auf die selbst erstellte Druckeranpassung abstimmen. Die Darstellung von Druckerzeichen, die nicht dem ATARI-Zeichensatz entsprechen, läßt sich dann noch über ein entsprechendes Programm zur Zeichensatzänderung bewerkstelligen, wie dies z.B. durch FONTMAKER möglich ist. Solche abzuändernden Zeichen dürfen dann selbstverständlich nicht ausgeblendet werden.

*ENDE*

# MACROMENIA
## oder
## "Als Vip-Professional das Laufen lernte"

**Wie der Titel schon sagt, werden wir unserem Vip-Professional heute gehörig Dampf machen und in die "Macrosprache" einsteigen. Vorab wiederum meinen besten Dank dem *"vierundvierzigsten"* Anrufer, der mich letzte Woche um 23.30h aus dem VIP-Schlaf gerissen und mir nahegelegt hat, etwas professioneller loszuschreiben. Doch vielleicht ist ihm die Einleitung im 1.Teil des Kurses noch nicht in die Hände gekommen, da stand nämlich "KURS für absolute Beginner". Seien Sie mir deshalb nicht böse, wenn ich auf die ca. 85 % Anfänger Rücksicht nehme und etwas für die Basis schreibe. Vielleicht habe ich auch mal was für Semiprofis im Repertoire, die in den Teilen 5 fff. bestimmt nicht zu kurz kommen werden.**

Den LOTUS-Anhängern, die mit 1-2-3 arbeiten und sich bei meinem Lektor Marcelo Merino ein klein wenig ausgesprochen haben, VIP sei nun mal nicht 1-2-3, denen sei bestätigt, daß ich genauso denke. Lotus 1-2-3 V1.A ist zwar identisch mit dem Befehls-, Formel-, und Syntaxaufbau, 1-2-3 ist jedoch um ein Vielfaches schneller (Textvip 8*langsamer, Gemvip ca. 10*langsamer) und auch leistungsfähiger, was die Speicherverwaltung anbelangt. (VIP schluckt fast 45 % mehr Speicherplatz)
1-2-3- Anwender, die mit der Version 2.01 arbeiten und dann im VIP schnuppern, werden sehr schnell bemerken, daß hier einiges fehlt.




## VIP-Version 2.0?

Doch nun ein Aufruf an alle! Wäre es nicht schön, wenn VIP-Professional, angenähert an die LOTUS 1-2-3 Version 2.01 oder 3.0, auch für uns, den ehrgeizigen und ernsthaften Atari-Anwender, zu haben wäre? Vielleicht kann uns da die Firma Kiekbusch helfen, die ja einen sehr guten Draht zu den heutigen Supportern von VIP-Professional hat ?

## Zeigt her Eure Hausaufgaben

Bevor die verzweifelten "VIP-Schüler" die Hausaufgabe des letzten Teils noch kurz in der großen Pause vom Nebensitzer abpinseln, zeige ich ihnen im Anhang die Auflösung (Lösungslisting). Bauen Sie bitte Ihr Arbeitsblatt wieder so auf, daß wir alle von einem gemeinsamen Ausgangspunkt starten können. Unsere ersten Macrosequenzen werden speziell für diese Anwendung geschrieben. Doch zuvor sollte ich vielleicht erklären, was ein Macro in VIP-Professional ist.

## MIDI, MINI, MACRO

Macros sind sequentiell, also nacheinander, ablaufende Befehlsfolgen, die nach einer Namensgebung in unserem VIP-Professional beliebige, zuvor festgelegte Tastenanschläge und/oder Befehle ausführen können. Das hört sich sehr kompliziert an, ist es aber überhaupt nicht.
Wir wollen unser Macroproblemchen mit einer etwas einfachen Anwendung angehen. Sie haben sicher alle bemerkt, daß beim Abspeichern und Laden unseres Arbeitsblattes auch die letzte Cursorposition abgespeichert wird. Stellen Sie den Cursor auf Zelle **AA1** und geben Sie unsere Tastatursequenz zum Speichern des Arbeitsblattes ein: /, **F**ile, **S**ave, (mit Return den Filenamen bestätigen), **R**eplace.
Laden Sie das Arbeitsblatt wieder ein: /, **F**ile, **R**etrieve. Sie sehen, die Cursorposition ist mit dem gesamten Bildschirmaufbau an der Zelle **AA1** orientiert. Dies kostet unnötig viel Bildschirmbewegung beim Rückladen der Anwendung und Rücksprung auf die Zelle **A1**.

| Klartext | Macrosyntax |
|---|---|
| Befehlsleiste | / |
| Worksheet | W |
| Titles | T |
| Clear | C |
| Taste:CLR-HOME | {HOME} |
| Taste:F5 {GOTO} | {GOTO} |
| C1 | C1 |
| Return | ~ |
| Befehlsleiste | / |
| Worksheet | W |
| Titles | T |
| Vertical | V |

Macrogenerierung:

/WTC{HOME}{GOTO}C1~/WTV

*Abb. 2: Macrosyntax*

Hätten wir nach dem Einladen der Anwendung die **CLR-Home-Taste** gedrückt, wäre der Bildschirm ideal aufgebaut. Mit etwas Raffinesse und VIP-Know-How können wir sogar ein Einfrieren von Spalten und Zeilen bewerkstelligen. Dieses Einfrieren erlaubt uns, später in der Anwendung zu marschieren, ohne die Zeilentitel und Spaltenüberschriften aus dem Blickfeld zu verlieren.

## AUTOSTART-MACRO

Die eleganteste Lösung für unser Rückladeproblem ist der AUTOSTART-Macro mit dem Namen **\0**. Dieser Macroname wird bei jedem Rückladen einer Anwendung, in der dieser Macro gespeichert wurde, selbständig ausgeführt.
Wir wollen diesen Macro zunächst logisch aufbauen und nehmen dazu ein Blatt Papier zur Hand. Tun Sie mal so als würden Sie meine Fingeranschläge nach Abbildung 2 gemäß der Macrosyntax notieren.

Alle Tastenanschläge erhalten ein Kürzel,das mit den Anfangsbuchstaben der Menüzeilen identisch ist, Funktions- und Sondertastenanschläge werden in spitze Klammern { } gesetzt und mit Klartext der Anweisung gefüllt. Ausnahmen gibt es hier auch, denn die Return-Taste muß mit dem Sonderzeichen ~ übergeben werden. Wie ich Sie bereits kenne, haben Sie fast alles verstanden und könnten Ihr Arbeitsblatt gemäß der Abbildung 3 aufbauen.
Bitte vergessen Sie nicht, die Zelleneintragungen **AD5**, **AD7** und **AD9** mit dem Sonderzeichen ‘ zu beginnen, denn VIP weiß ja nicht, ob Sie die Befehlsleiste rufen oder ein Macro generieren möchten.

Wenn alle Eintragungen erfolgreich abgeschlossen sind, positionieren Sie den Cursor auf Zelle **AD5** und benennen Sie Ihren Macro mit der Taufprozedur: /, **R**ange, **N**ame, **C**reate, **\0**, (Return), (Return) Bitte Cursor auf **AD7**, Taufglocken läuten: /, **R**ange, **N**ame, **C**reate, **\S**, (Return), (Return). Bei Zelle **AD9** verwenden Sie nur den Namen **\D**. Bis hierher war es ja ganz einfach, doch die PD-Diskbesitzer Nr.131 wissen, daß MACROS schnell und gefääääâhhhrlich sind. Bitte speichern Sie Ihr Arbeitsblatt deshalb nach jeder Macrogenerierung ab, Sicherheitsfanatiker legen sich sogar noch eine Kopie Ihres Arbeitsblattes an. (Letzterem Aufruf sollten Sie bei der Arbeitsblattentwicklung Folge leisten).

## Ich hab's doch gewußt!

Schlauberger unter Ihnen werden die Taufsequenz zur Namensgebung selbstverständlich sofort in den Macro \C verwandeln. Hier die Lösung: **/RNC{?}~~/ xq~** ( **B**efehlsleiste, **R**ange, **N**ame, **C**reate, (Warten auf den Namen), Return, Return, Ende des Macros).

## MERLIN zückt den Macrozauberstab

Wer kennt Ihn nicht den großen Zauberer Merlin, der in der Sagenwelt lebt, und dem man Phantastisches nachsagen kann. Anscheinend hat er jetzt auch unserem VIP-Arbeitsblatt etwas Zauberpulver in die Zellen gestreut. Beim Rückladen unserer Anwendung passiert nämlich etwas Wundersames. Das Arbeitsblatt wird aufgebaut, der Cursor springt blitzschnell auf C1, und bewegen wir uns gen Osten, scheint der gesamte Westen (damit meine ich die Zeilenbeschriftungen) eingefroren zu sein. Niemand anderes als der Macro **\0** ist an diesem Zauberstückchen schuld.
Das Abspeichern unseres Arbeitsblattes kann jetzt auch mit dem direkten Macroaufruf: TASTE **ALT+S** ausgeführt werden. Hat doch ganz prima geklappt! Nun noch unser Ausdruck (auf IBM-/EPSON-kompatiblen Druckern) mit dem Macroaufruf: TASTE **ALT+D**.

## MEHR, MEHR, MEHR......

habe ich 1983 gesagt, als Lotus 1-2-3 in unserem Unternehmen eingeführt wurde und ich nach 3-monatigem Handbuch- und Programmstudium (alles in Englisch) wegen Ernährungs und Schlafmangel von meiner MS-DOS-Kiste weggezerrt wurde und zur Erholung einige Wochen Familie verordnet bekommen habe. Macros sind stark und vielfältig. Einmal dieser Macrosucht verfallen- und Sie kommen davon nicht mehr los. Haben Sie Ihr Arbeitsblatt einmal mit Macros ausgestattet, dürfen Sie es wie es die Profis tun, ab sofort **Template** nennen.
Damit es Ihnen leichter fällt, alle Tastatur- und Funktionstastenanschläge im Macrofieber auch in Ihre Zellen zu kriegen, zeigt die Abbildung 4 zeigt eine vollständige Macroreferenzliste:

| | AA | A | AC | AD |
|---|---|---|---|---|
| 1 | Macros: | | | |
| 2 | ------- | | ------- | ------- |
| 3 | Ausführung | | Name | Syntax |
| 4 | ------- | | ------- | ------- |
| 5 | AUTOSTART | | \0 | /wtc{HOME}{GOTO}C1~/wtv |
| 6 | | | | |
| 7 | SICHERN | | \S | /fdA:\~/FS~R/xq~ |
| 8 | | | | |
| 9 | DRUCK | | \D | /PPCARA1.E11~ |
| 10 | | | | OS\015~P72~ |
| 11 | | | | ML4~MR80~ |
| 12 | | | | MT4~MB4~ |
| 13 | | | | HDarlehnsvergleiche: vom @~ |
| 14 | | | | QAGQ/XQ~ |

Macro: \0

```
Befehlsleiste
Worksheet
Titles
Clear
{CLR-HOME}
{F5}
C1
RETURN
Befehlsleiste
Worksheet
Titles
Vertical
```

Macro: \S

```
Befehlsleiste
File
Directory
A:\
Return
Befehlsleiste
File
Save
Replace
Macroende mit
          XQ
```

Macro: \D

```
Befehlsleiste   | Options     | Margin Left  | Margin Bottom
Print           | Setup       | 4            | 4
Printer         | \015        | Return       | Return
Clear           | Return      | Margin Right | Header
All             | Pagelength  | 80           | Darlehnsvergleiche:von @(Datum)
Range           | 72          | Return       | Return
A1.E11          | Return      | Margin Top   | Quit
Return          |             | 4            | Align
                |             | Return       | GO
                |             |              | Quit und Macroende mit XQ
```

*Abb. 3: Macros in unserer Anwendung.*

| Taste/Operation | Macrosyntax |
|---|---|
| Befehlsleiste | / |
| Cursor-Nord | {UP} |
| Cursor-Süd | {DOWN} |
| Cursor-Ost | {LEFT} |
| Cursor-West | {RIGHT} |
| CLR-HOME | {HOME} |
| Insert | {INS} |
| Backspace | {BS} |
| Return | ~ |
| Delete | {DEL} |
| Shift+Cur.Nord | {PGUP} |
| Shift+Cur.Süd | {PGDN} |
| .. END /Nord/Süd | {END}... |

| Funktionstasten | | |
|---|---|---|
| F 2 | Edit | {EDIT} |
| F 3 | Name | {NAME} |
| F 4 | Abs | {ABS} |
| F 5 | Goto | {GOTO} |
| F 6 | Window | {WINDOW} |
| F 7 | Query | {QUERY} |
| F 8 | Table | {TABLE} |
| F 9 | Calc | {CALC} |
| F 10 | Graph | {GRAPH} |

| Sonderfunktion | |
|---|---|
| Warten auf Eingabe | {?} |

*Abb. 4: Macroreferenzliste*

KURS3_2

| | AA | A | AC | AD | AE | AF | AG |
|---|---|---|---|---|---|---|---|
| 5 | AUTOSTART | | \0 | /wtc{HOME}{GOTO}C1~/wtv/xgMENUE~ | | | |
| 6 | | | | | | | |
| 7 | MENUE | | MENUE | /xmAD8~ | | | |
| 8 | | | \M | Eingabe | Druck | Sichern | Quit |
| 9 | | | | Daten eingeben | drucken | A:\ sichern | Ende |
| 10 | | | | /wtc{HOME} | /xg\D~ | /xg\S~ | /xq~ |
| 11 | | | | {GOTO}C1~/wtv/xq~ | | | |
| 12 | | | | | | | |
| 13 | SICHERN | | \S | /fdA:\~/FS~R/xq~ | | | |
| 14 | | | | | | | |
| 15 | DRUCK | | \D | /PPCARA1.E11~ | | | |
| 16 | | | | OS\015~P72~ML4~MR80~MT4~MB4~ | | | |
| 17 | | | | HDarlehnsvergleiche: vom @~ | | | |
| 18 | | | | QAGQ/XQ~ | | | |
| 19 | | | | | | | |

*Abb 5: Menümarcos in unserer Anwendung.*

## War das schon alles, lieber Merlin?

Denkste, jetzt geht's erst richtig los! Die Befehlsleiste mit ihren Untermenüaufrufen ist ja wirklich eine tolle, wenn auch anfangs etwas verwirrende Angelegenheit. Wir wollen in unser Template eine ebenso elegante Menüsteuerung integrieren.

Bitte bauen Sie Ihr Arbeitsblatt nach Abbildung 5 auf:

Mit unserem Taufmacro **\C** müssen wir wieder 2 Macronamen generieren bzw. benennen. Cursor auf **AD6** stellen: **/RNCMENUE~~** oder **ALT+C** (Taufmacro) und den Namen **MENUE** eingeben. Das gleiche nochmal: **/RNC\M~~** oder **ALT+C** (Taufmacro) und Namen **\M** eingeben. Bitte das Sichern nicht vergessen und jetzt Augen zu, **ALT+M** gedrückt, staunen, wundern und sich freuen, unser erstes Menü ist fertig.

## Menüs kann man nicht essen!

Aber elegant sind sie schon, nehmen eine Menge Tipperei ab und selbst mein 2-jähriger Sohn könnte das Darlehens-Template bedienen, wenn er nur schon lesen könnte.

In dem Menüaufbau traten erstmals die **/x-Kommandos** auf, die es ermöglicht haben, Sprungziele innerhalb eines Menüs zu gestalten, sowie eine Menüleiste in der VIP-Shell einzublenden. In Abbildung 6 erhalten Sie die Gesamtübersicht der **/x-Kommandos**.

## Do you speak VIP?

Ja, ab jetzt sprechen wir in einer anderen Sprache! Die nächtlichen Träume werden von den Basic-, Pascal- und C-Befehlen auf die Vip-Sprache abwandern. Man wird so manche Nacht mit einem Template zu kämpfen haben, um Macros einen sinnvollen Aufbau zu geben. Mit Erstaunen habe ich letzte Woche erfahren, daß alle neuen Mega-STs mit diesem phantastischen VIP-Professional ausgeliefert werden (ohne Aufpreis). Ich kann nur hoffen, daß man auch an die US-TEXT und US-GEM-Versionen gedacht hat, denn wie wir alle bereits festgestellt haben, werden die Macros innerhalb einer US-Shell nicht in der DEUTSCHEN-Shell lauffähig sein, geschweige denn

| /x-Kommandos | Funktion |
|---|---|
| **Sprünge** | |
| /XG (Koordinate/Macroname) | = GOTO |
| /XC (Koordinate/Macroname) | = GOSUB |
| /XR | = RETURN |
| **Menuedeklaration** | |
| /XM (Bereich/Name) | = Menue |
| **Eingabeaufforderungen** | |
| /XL (Meldung)~(Koordinate) | = Texteingabe |
| /XN (Meldung)~(Koordinate) | = Zahleneingabe |
| **Bedingte Abfrage** | |
| /XI (Bedingung) | = IF/THEN |
| **Ende** | |
| /XQ | = Macroende |

*Abb. 6: /X-Kommandos*

wird mit der DEUTSCHEN-Shell eine Macrokompatibilität zu den großen Meistern Lotus 1-2-3 oder Symphony möglich sein.
Seltsamerweise (tut mir leid wenn ich mich wiederholen muß) ist im deutschen VIP (1.6) der Menüaufbau in deutsch, die Formeln und /X-Kommandos in englisch implementiert. Ein Versionsaustausch deutsch/englisch ist also mit Schwierigkeiten verbunden. Die universellste Version ist und bleibt die US-Shell 1.2 - 1.4 GEM/TEXT. Nach meinen Informationen stellen die Kieckbusch-Mannen gegen UPDATE-Gebühr auch noch diese Version zur Verfügung. Eine weitere erfreuliche Nachricht kann ich allen "Sicherheitsfreundschaftskopienbesitzern" machen: Es gibt aus Überproduktionsbeständen für jedermann ein VIP-Professional Handbuch (50,- DM) zu kaufen.
Es lohnt sich bestimmt ein solches Handbuch nachträglich zu erwerben. Wenn der didaktische Aufbau auch sehr an eine Direktübersetzung erinnert, erhalten Sie jedoch eine umfangreiche Übersicht aller Funktionen, Besonderheiten und Formeln. Den heißgemachten und SEMIs kann ich auch ein weiteres Nachschlagewerk empfehlen. Es ist zwar für Lotus 1-2-3 V 1.A geschrieben, diente mir aber im Anfangsstadium der Macromenia als sinnvolles Nachschlagewerk: *"Das Lotus 1-2-3- Kompendium", Verlag Markt und Technik, Autoren Geoffrey T. LeBlond und Douglas Ford Cobb*, übersetzt von *Kurt Hinze und Dr. Horst Schmalfeld. ISBN:3-89090-005-4* (Preis ca. 58,— DM)

## Ein VIP-FORUM?

Warum nicht, sprach Marcelo Merino, als wir über die vielen Briefe, Hilferufe und wirklich ernstzunehmenden Anrufe (43) diskutiert haben. Wir möchten alle VIPs gerne auffordern Ihre Lösungsvorschläge, Problemchen, Tips, Tricks und Kniffe allen VIP-Kämpfern zugänglich zu machen. Ihre Vorschläge und Problemlösungen werden, natürlich Ihr Einverständnis vorausgesetzt, in der ST veröffentlicht und besprochen. So könnten wir dem Bilanz- und Lohnbuchhalter, Vereinskassier, Versicherungsmakler, Zahnarzt, Hochschulprofessor und der Hausfrau helfen, ihre Anwendungen mit VIP zu erledigen.

Also greifen Sie zum Atari-Ferrari, schmeißen Sie das VIP-Template rein, merzen Sie alle Fehler aus, schreiben Sie eine kleine Erklärung dazu und schicken Sie das ganze an Merlin (nicht zu verwechseln mit dem Zauberer) oder direkt an mich, ich werde das Template durch den kritischen Testfilter meiner konstanten VIP-Viren jagen und dann nach Rücksprache gerne allen zur Verfügung stellen.

| Operator | Operation | Reihenfolge |
|---|---|---|
| ^ | Potenzierung | 7 |
| -+ | negativ, positiv | 6 |
| */ | Multiplikation/Division | 5 |
| +- | Addition, Subtraktion | 4 |
| =<> | gleich, ungleich | 3 |
| <> | kleiner als, größer als | 3 |
| < = | kleiner als oder gleich | 3 |
| > = | größer als oder gleich | 3 |
| #NOT# | logisches NICHT | 2 |
| #AND# | logisches UND | 1 |
| #OR# | logisches ODER | 1 |

*Abb. 7: Operatoren*

```
478 + ((5800 + C8) * 0.1809) / 12 - C9
    |        |     |              |   |
    4        1     2              3   5
```

*Abb. 8: Reihenfolgen*

## Rückblende und Versprechen

In Teil 2 habe ich versprochen, noch etwas aus dem Formelsprachschatz zu plaudern. Da VIP nun allen neuen MEGA-ST-Käufern zugänglich gemacht wird, kann ich auch ohne vorgehaltene Hand und böse Blicke der Atari- und Kiekbusch-Mannen noch etwas aus dem Formelsprachschatz rezitieren. Beginnen wir mit den **OPERATOREN** nach Abbildung 7.

Mit der Reihenfolgenummer wird die Reihenfolge angegeben, in der VIP Operationen in einer Formel ausführt. VIP führt die Operation mit einer höheren Reihenfolgenummer vor einer Operation mit einer niedrigeren Reihenfolgenummer aus. Operationen mit derselben Reihenfolgenummer werden nacheinander von links nach rechts ausgeführt.
Reihenfolgenummern können außer Kraft gesetzt werden, indem eine Operation in Klammern gesetzt wird. VIP führt Operationen in Klammern als erstes aus. Innerhalb der Klammerpaare gelten die Reihenfolgenummern. In einer Formel können bis zu 32 Klammerpaare benutzt werden.
Das folgende Beispiel in Abbildung 8 zeigt, wie VIP die Operationen nach Reihenfolgenummer ausführt:

| Funktion | Kurzbeschreibung |
|---|---|
| @ABS | absoluter Wert einer Zahl |
| @EXP | Wert der Konstante "e" |
| @INT | ganzzahliger Anteil |
| @LN | natürlicher Logarithmus "e" |
| @LOG | Logarithmus Basis 10 |
| @SQRT | Quadratwurzel |
| @RAND | Zufallszahl |
| @ROUND | Rundungen mit Nachkommastellen |
| @MOD | Anzeige: Rest einer Division |

*Abb. 9: Mathematik*

# Reinhard Schuster Computer

**OBERE MÜNSTERSTR. 33–35 · TEL. (0 23 05) 37 70 · BTX 023053770 · 4620 CASTROP-RAUXEL**

Schneider COMPUTER DIVISION Vertragshändler · ATARI System-Fachhändler · star der ComputerDrucker Vertragshändler · AMSTRAD Vertragshändler · Commodore Vertrags-Werkstatt

## ATARI-ST-SOFTWARE

| Titel | Preis |
|---|---|
| 10Th Frame | 66.60 |
| 221B Baker Street | 47.90 |
| 3D Galaxy | 54.10 |
| 500 CC Grand Prix | 58.80 |
| Addicta Ball | 47.90 |
| Advanced Art Studio | 76.60 |
| Airball | 76.60 |
| Airball Construction Set | 47.90 |
| Altair | 54.10 |
| Alternate Reality City | 76.60 |
| Amazon | 54.10 |
| American Pool | 29.30 |
| Annals of Rome | 74.30 |
| Arcade Force Four | 69.90 |
| Arena | 88.30 |
| Arena/Braticass | 89.90 |
| Arkanoid | 38.60 |
| Asphai Trilogy | 76.60 |
| Asterix im Morgenland | 59.90 |
| Autoduel | 58.80 |
| Backlash | 49.90 |
| Bad Cat | 55.70 |
| Balance of Power | 85.20 |
| Barbarian | 47.90 |
| Bard's Tale 1 | 79.90 |
| Baseball Gamestar | 76.60 |
| Battleships | 47.90 |
| Beyond Zark | 74.30 |
| Black Cauldron | 76.60 |
| Black Lamp | 58.80 |
| Blue War | 54.10 |
| Blueberry und das Gespenst | 59.90 |
| Bob Moran in Middle Age | 59.90 |
| Bob Morane Sience Fiction | 58.80 |
| Borrowed Time | 59.90 |
| Boulderdash Construction Set | 59.90 |
| Brataccas | 105.40 |
| Brian Clough Football | 76.60 |
| Bubble Bobble | 58.80 |
| Bubble Ghost | 58.80 |
| Bureaucrazy | 105.40 |
| Captain America | 58.80 |
| Carrier Command | 58.80 |
| Chamonix Challenge | 58.80 |
| Championship Baseball | 69.90 |
| Championship Football | 76.60 |
| Championship Wrestling | 76.60 |
| Checkmate | 29.90 |
| Chopper X | 27.80 |
| Clever & Smart | 58.80 |
| Colonial Conquest | 89.90 |
| Crafton & Xunk | 59.90 |
| Crash Garet | 59.90 |
| Crazy Cars | 55.70 |
| Creator | 59.90 |
| Crystal Castles | 47.90 |
| Cutthroats | 89.90 |
| Dark Castle | 69.90 |
| Deathtrike | 33.60 |
| Deep Space | 105.40 |
| Defender of the Crown | 77.40 |
| Deja Vu | 66.60 |
| Diablo | 59.90 |
| Dizzy Wizzard | 55.70 |
| Dungeon Master | 74.30 |
| Eco | 58.80 |
| Eden Blues | 69.90 |
| Electronic Pool | 57.20 |
| Enduro Racer | 58.80 |
| Epyx (The Worlds Greatest) | 77.40 |
| Extensor | 59.90 |
| Extravaganza | 35.50 |
| Eye | 43.30 |
| Fahrenheit 451 | 54.10 |
| Fire Blaster | 29.90 |
| Flight Sim. Scenery Disk 11 | 57.20 |
| Flight Sim. Scenery Disk 7 | 57.20 |
| Flight Simulator 2 | 138.00 |
| Football Fortunes | 69.90 |
| Football Manager | 43.30 |
| Fred Feuerstein | 58.80 |
| Fright Night | 55.70 |
| Frost Byte | 47.90 |
| Gambler | 38.60 |
| Gato | 85.20 |
| Gauntlet 1 | 69.90 |
| Gountlet 2 | 69.90 |
| Gnome Ranger | 43.30 |
| Gold Runner | 69.90 |
| Golden Path | 57.20 |
| Guild of Thieves | 69.90 |
| Gunship | 74.30 |
| Hacker | 59.90 |
| Hacker 2 | 69.90 |
| Hardball | 66.60 |
| Harrier Strike Mission | 88.30 |
| Hellowoon | 69.70 |
| Hitchhikers Guide to Galaxy | 88.30 |
| Hollywood Hi Jinx | 88.30 |
| Hollywood Poker | 47.90 |
| Hot Ball | 69.90 |
| Impact | 43.30 |
| Indiana Jones | 49.90 |
| Infidel | 89.90 |
| International Karate | 59.90 |
| Into the Eagles Nest | 58.80 |
| Isnogud | 58.80 |
| Jagd auf Roter Oktober | 74.30 |
| Jewels of Darkness | 59.90 |
| Jinxter | 74.30 |
| Joe Blade | 29.30 |
| Jump Jet | 43.30 |
| Jupiter Probe | 43.30 |
| Karate Kid 2 | 66.60 |
| Karate Master | 39.90 |
| Karting Grand Prix | 29.30 |
| Kings Quest 3er Pack | 76.60 |
| Knight Orc | 58.80 |
| L'Affaire | 76.60 |
| Las Vegas | 33.60 |
| Leaderboard Golf | 69.90 |
| Leaderbord Tournament | 33.60 |
| Leather Goddesses of Phobos | 79.90 |
| Leatherneck | 55.70 |
| Leisure Suit Larry | 59.90 |
| Leviathan | 43.30 |
| Liberator | 39.90 |
| Little Computer People | 105.40 |
| Livingstone | 47.90 |
| Luky Luke | |
| Nitroglyzerin | 57.20 |
| Lurking Horror | 89.90 |
| MacAdam Bumper | 59.90 |
| Mach 3 | 58.80 |
| Marble Madness | 83.60 |
| Mercenary | 66.60 |
| Metrocross | 66.60 |
| Metropolis | 35.50 |
| Mewilo | 66.60 |
| MGT | 69.90 |
| Microleague Wrestling | 69.90 |
| Mindshadow | 58.80 |
| Missing One Droid | 33.60 |
| Mission Elevator | 58.80 |
| Moebius | 74.30 |
| Moonmist | 89.90 |
| Mortville Manor | 76.60 |
| Mouse Trap | 47.90 |
| Music Studio | 105.40 |
| Nine Princess in Amber | 54.10 |
| Ninja Mission | 29.90 |
| Nord & Bert | 76.60 |
| Not a Penny More | 59.90 |
| Obliterator | 77.40 |
| Ogre | 74.30 |
| Oids | 58.80 |
| Outcast | 29.90 |
| Pacland | 58.80 |
| Passengers on the Wind | 59.90 |
| Passengers on the Wind 2 | 59.90 |
| Perfect Match | 33.60 |
| Perry Mason | 54.10 |
| Phantasie 2 | 76.60 |
| Phantasie 3 | 66.60 |
| Phoenix | 58.80 |
| Pinball Factory | 66.60 |
| Pink Panther | 58.80 |
| Pirates of the Barbery Coast | 35.50 |
| Planetfall | 89.90 |
| Plundered Hearts | 76.60 |
| Plutos | 43.30 |
| Police Quest | 58.80 |
| Pool | 29.30 |
| Pool/Shuffle Board | 47.90 |
| Predator | 58.80 |
| Protector | 29.90 |
| Psion Chess | 76.60 |
| Q Ball | 57.20 |
| Quantum Paint Box | 59.90 |
| Rampage | 47.90 |
| Rana Rama | 58.80 |
| Return to Genesis | 58.80 |
| Rings of Zilfin | 66.60 |
| Roadrunner | 66.60 |
| Roadwar 2000 | 76.60 |
| Roadwar Europe | 66.60 |
| Rockford | 58.80 |
| Rogue | 76.60 |
| Rolling Thunder | 55.70 |
| Sapiens | 58.80 |
| Seastalker | 89.90 |
| Seconds Out | 55.70 |
| Sentinel | 58.80 |
| Shanghai | 69.90 |
| Shuffleboard | 29.30 |
| Shuttle 2 | 66.60 |
| Sidewalk | 58.80 |
| Silent Service | 69.90 |
| Silicon Dreams | 59.90 |
| Sky Blaster | 58.80 |
| Sky Fighter | 43.30 |
| Suspect | 89.90 |
| Swooper | 62.30 |
| T.N.T. | 52.60 |
| Taipan | 49.90 |
| Tanglewood | 54.10 |
| Tass Times | 69.90 |
| Tee up Golf | 43.30 |
| Terramax | 58.80 |
| Terrorpods | 66.60 |
| Tetris | 54.10 |
| Thai Boxing | 43.30 |
| The Pawn | 69.90 |
| Thrust | 29.90 |
| Time Bandit | 89.90 |
| Time Blast | 33.60 |
| Tracker | 74.30 |
| Trailblazer | 63.50 |
| Trasheap | 58.80 |
| Trauma | 59.90 |
| Trinity | 105.40 |
| Trivial Genus Edit. | 59.90 |

**Telefonische Bestellung:**

**(02305) 3770**
**(Tag und Nacht)**

| Titel | Preis |
|---|---|
| Skyrider | 58.80 |
| Slaygon Adventure | 55.70 |
| Solomons Key | 54.10 |
| Space Ace | 58.80 |
| Space Baller | 29.30 |
| Space Pilot | 47.90 |
| Space Port | 58.80 |
| Space Quest 1 | 76.60 |
| Space Quest 2 | 58.80 |
| Space Station | 46.40 |
| Spiderman | 59.90 |
| Spy Versus Spy | 66.60 |
| ST Classics | 59.90 |
| ST-Soccer | 55.70 |
| Star Raiders | 43.30 |
| Starglider | 69.90 |
| Startrek | 58.80 |
| Starwars | 58.80 |
| Stationfall | 88.30 |
| Strike | 33.60 |
| Strike Force Harrier | 69.90 |
| Strip Poker | 59.90 |
| Strip Poker 2 | 43.30 |
| Sub Battle Simulator | 66.60 |
| Super Cycle | 76.60 |
| Super Huey | 59.90 |
| Super Sprint | 43.30 |
| Superstar Icehockey | 69.90 |
| Triviar Trove | 27.80 |
| Turbo | 27.80 |
| Turbo GT | 49.90 |
| Turbo ST | 33.60 |
| Two on Two Basketball | 76.60 |
| Typhoon | 59.90 |
| Ultima 2 | 89.90 |
| Ultima 3 | 69.90 |
| Ultima 4 | 74.30 |
| Universal Military Sim. | 74.30 |
| Vegas Gambler | 49.90 |
| Vermeer | 77.40 |
| War Games Contr. Set | 66.60 |
| War Hawk | 29.90 |
| Warlock's Quest | 43.30 |
| Warzone | 29.90 |
| Waterskiing | 58.80 |
| Western Games | 58.80 |
| Winter Olympiad '88 | 55.70 |
| Wintergames | 69.90 |
| Wishbringer | 89.90 |
| Wizball | 58.80 |
| Wizzards Crown | 76.60 |
| Xenon | 58.80 |
| Xevious | 66.60 |
| Zork 1 | 89.80 |

## NEUHEITEN

| Titel | Preis |
|---|---|
| BMX Simulator | 39.90 |
| Soccer Sopremo | 39.90 |
| AB Zoo | 47,90 |
| Bob Winner | 67,00 |
| Bobo | 58,80 |
| Bridge Player 2000 | 59,90 |
| Buggy Boy | 58,80 |
| Captain Blood | 69,90 |
| Casino Roulette | 59,90 |
| Deflector | 59,90 |
| Footballmanager 2 | 58,80 |
| Formula 1 Gr. Pr. Sim. | 55,70 |
| Foundations Waste | 69,90 |
| Get Dexter 2 | 69,90 |
| Gold Runner 2 | 58,80 |
| Gold Run. 2 Sc. Disc 1 | 18,50 |
| Gold Run. 2 Sc. Disc 2 | 18,50 |
| Hollywood Poker | 36,40 |
| Hot Ball | 69,90 |
| I Ball | 33,60 |
| Ikari Warrior | 47,90 |
| Impossible Mission 2 | 55,70 |
| Indoor Sports | 69,90 |
| Kaiser | 129,90 |
| Legends of the Sword | 69,90 |
| North Star | 59,90 |
| Out Run | 58,80 |
| Pengy | 47,90 |
| Powerplay | 59,90 |
| Rimrunner | 47,90 |
| ST Wars | 69,90 |
| Scruples | 59,90 |
| Spitfire 40 | 69,90 |
| Storm Trooper | 58,80 |
| Test Drive | 79,90 |
| Vampires Empire | 58,80 |

## HARDWARE

**Unser Superknüller**
**Akustikkoppler Hitrans 300 P**

300 Baud, vollduplex, RS 232 C-Schnittstelle, flexibles Mittelteil, Stromversorgung über Batterie, Akku oder mitgeliefertem Netzteil möglich. Incl. FTZ-Nr. (Postzulassung), Handbuch und Netzteil.

**198,–**

**Kunstlederhauben**

| Artikel | Preis |
|---|---|
| 260 /520 ST | 17.90 |
| 520/1040 STF | 21.90 |
| Mega ST Keyboard | 21.90 |
| Mega ST Keyboard/SM 124 | 46.90 |
| Mega Keyboard/SM 125 | 48.90 |
| Floppy 314/354 | 14.90 |
| Monitor SM 124 | 27.90 |
| Monitor SM 125 | 29.80 |
| Monitor SC 1224 | 32.90 |

**Zweitlaufwerk** 3½'', 720 KB mit Netzteil **348,–**

**Markendisketten:**

| Artikel | Preis |
|---|---|
| 3½'' 1D | 24.90 |
| 3½''-2 D | 29.80 |
| Mouse-Pad | 19,80 |
| Floppy-Stecker, 14-pol. | 7,90 |
| Floppy-Kupplung 14-pol. | 7.90 |
| Monitor-Stecker 13-pol. | 7,90 |
| Monitor-Kuppl. 13-pol. | 7,90 |

**Diskettenbox 3 + 3½''**
für 80 3½ Disketten, abschließbar

**17.90**

**Ladengeschäftszeiten:**
**Montag–Freitag 9.00 – 13.00 Uhr**
**15.00 – 18.30 Uhr**
**Samstag 9.00 – 14.00 Uhr**
**Langer Samstag 9.00 – 18.00 Uhr**

**Versand** per Nachnahme zuzügl. Versandkosten. Oder Vorkasse auf Psch.-Kto. Nr. 69422-460 PschA Dortmund zuzügl. 5,– DM Versandkosten.
Ausland nur per Vorkasse auf Psch.-Kto. zuzügl. 10,– DM Versandkosten. **Bitte bei allen Bestellungen Computertyp angeben!**

**Besuchen Sie unser Ladengeschäft und lassen Sie sich durch unser Fachpersonal beraten. Wir haben laufend günstige Angebote und stark reduzierte Vorführgeräte.**

○ Senden Sie mir bitte Ihren Katalog (2,– DM in Briefmarken liegen bei)
○ Hiermit bestelle ich per Nachnahme:
○ Incl. kostenlosem Katalog

Vorname, Name

Straße, Hausnummer

PLZ, Ort

Datum, Unterschrift

**BESTELLSCHEIN**

| Anz. | Artikel | Preis |
|---|---|---|
| | | |

# MATHEMATIK, oder das notwendige Übel!

### Beispiele:

**@ABS**

berechnet den absoluten Wert einer Zahl oder eines Zelleninhalts:

@ABS(-8) = 8
@ABS(-789) = 789
@ABS(3) = 3

**@EXP**

berechnet den Wert der Konstanten "e" (angenähert am 2.7182818) mit dem Exponenten, der durch den Klammerausdruck bestimmt ist.

@EXP(5) = 148.41316
@EXP(C8) = 15.642632 (C8=2.75)

Falls die Zahl oder der Zelleninhalt größer als 230 ist, erscheint die Fehlernachricht "ERR" (Error: englisch Fehler).

**@INT**

berechnet den ganzzahligen Anteil (INTEGER) der Zahl oder des Zelleninhalts.

@INT(8.987679) = 8
@INT($C$7) = 78 (wenn C7= 78.9878867 ist)

Zu beachten ist, daß @INT im Gegensatz zur @ROUND-Funktion einfach alle Dezimalstellen abschneidet.

**@LN**

berechnet den natürlichen Logarithmus (zur Basis "e") der Zahl oder des Zelleninhalts.

@LN(17.634) = 2.8698289

Die Zahl oder der Zelleninhalt müssen positiv sein, sonst erscheint die Fehlermeldung "ERR". Bitte beachten Sie, daß die @LN-Funktion die Umkehrfunktion von @EXP ist.

**@LOG**

berechnet den Logarithmus (zur Basis 10) oder des Zelleninhalts.

@LOG(4.56) = 0.658964
@LOG(AB2) = 0.550839,

**@SQRT**

berechnet die Quadratwurzel der Zahl oder des Zelleninhalts.

@SQRT(5) = 2.236067
@SQRT(Z1) = 4, wenn Z1=16 ist.

**@RAND**

Die eingebaute @RAND-Funktion erzeugt Zufallszahlen zwischen 0 und 1 mit bis zu acht Dezimalstellen. Wenn Sie diese Funktion in eine Zelle eingeben, wird diese Zelle jedesmal einen anderen Wert zwischen 0 und 1 anzeigen, wenn die Tabelle neu berechnet wird.

**@ROUND**

Die @ROUND-Funktion wird benutzt, um Zahlen auf eine bestimmte Genauigkeit zu runden. Das allgemeine Format dieser Funktion ist:

**@ROUND(ZAHL,DEZIMALSTELLEN)**

Die Dezimalstellen können einen Wert zwischen 15 und -15 interpretieren und die Stellen rechts beziehungsweise links vom Dezimalpunkt wiedergeben.

@ROUND(19.98789,2) = 19.99
@ROUND(123.453,-2) = 100
@ROUND(123.456,-1) = 120

Der entscheidende Vorteil der @ROUND-Funktion ist, daß Sie den Rundungsfehler bei Geldbeträgen verringern. In unserer Darlehensanwendung ließe sich somit die 3. Nachkommastelle eliminieren.

**@MOD**

Die @MOD- Funktion zeigt den Rest einer Division an. Das allgemeine Format ist (Modulo-Rechnung).

@MOD(Divident,Divisor)

@MOD(7,3) = 1 (7:3 = 2 Rest 1)
@MOD(71.3,21) = 8.3 (71,3:21=3 Rest 8.3)

Eine Nulldivison wird mit Fehlermeldung ERR ausgegeben.

# GEOMETRIE, Laster und Leiden!

VIP besitzt auch einen vollständigen Satz trigonometrischer Funktionen, die ich als Schüler immer verabscheut habe und bei denen ich nur durch Abschreiben bei meinen Mitschülern brillieren konnte. (Hoffentlich bekommt keiner meiner ehemaligen Professoren diesen Bericht in die Hände)

Viele VIP-Benutzer werden diese Funktionen sicherlich nie anwenden. Aber diejenigen, die VIP zum Lösen von Ingenieuraufgaben einsetzen, werden diese Funktionen nach Abbildung 10 nützlich finden.

**@PI**

gibt die Zahl PI(3.141592653589794) zurück. PI ist die Verhältniszahl eines Kreisumfangs zum Durchmesser.

@SIN(30*@PI/180) = 0.5
@PI*4^2 = 50.26548 der Inhalt eines Kreises Radius 4

**@SIN**

gibt den Sinus einer Zahl oder Zellenadresse zurück. Der Winkel (ZAHL) muß als Radiant ausgedrückt sein.

@SIN(@PI/6) = 0.5
@SIN(90*@PI/180) = 1
@SIN(30*@PI/180) = 0.5

**@COS**

gibt den Cosinus von Winkel x1 zurück. Das Ergebnis liegt zwischen -1 und 1 einschließlich. Der Winkel x muß als Radiant ausgedrückt werden.

@COS(60*@PI/180) = 0.5

**@TAN**

gibt den Tangens von Winkel x zurück. Der Winkel x muß in Radiant ausgedrückt werden.

@TAN(30*@PI/180) = 0.5

**@ASIN**

berechnet den Arkussinus eines Winkels. Diese Funktion gibt den Winkel als Radiant zurück, dessen Sinus gleich x ist. Das Ergebnis liegt stets zwischen -PI/1 und PI/2 und stellt einen Winkel im I. oder II.Quadranten dar. Das Argument x muß zwischen -1 und 1 einschließlich liegen.

@ASIN(-0.264) = -0.24855 (Radiant)
@ASIN(WINKEL) = 0.523598 (Radiant), wenn die Zelle namens WINKEL den Wert 0.5 enthält.

| Funktion | Kurzbeschreibung |
|---|---|
| @PI | Konstante PI (3.141592653589.) |
| @SIN | Sinusfunktion |
| @COS | Cosinusfunktion |
| @TAN | Tangensfunktion |
| @ASIN | Arkussinus |
| @ACOS | Arkuscosinus |
| @ATAN | Arkustangens |

*Abb. 10: Geometrie*

**@ACOS**

berechnet den Arkuscosinus eines Winkels. Diese Funktion gibt den Winkel als Radiant zurück, dessen Cosinus gleich xl ist. Das Ergebnis liegt stets zwischen 0 und PI und stellt den Winkel im I. oder II.Quadranten dar. Das Argument x muß zwischen -1 und 1 einschließlich liegen.

@ACOS(-0.3) = 1.875488 (Radiant)

**@ATAN**

berechnet den Arkustangens im III.Qadranten eines Winkels. Der Winkel, dessen Tangens ZAHL ist, wird als Radiant zurückgegeben. Das Ergebnis liegt stets zwischen -PI/2 und PI/2 und stellt einen Winkel im I. oder IV. Quadranten dar. Für den Wert des Arguments x gibt es keine Einschränkungen.

@ATAN(1) = 0.785398 (Radiant)

## Finanzmathematische Funktionen oder der Weg zum Geld!

| Funktion | Kurzbeschreibung |
|---|---|
| @IRR | Interner Zinsfuß |
| @NPV | Barwertsberechnung |
| @FV | Endwert einer Annuität |
| @PV | Barwert einer Annuität |
| @PMT | gleichbleibende Rückzahlungsraten pro Periode für ein Darlehn |

*Abb. 11: Finanzmathematik*

**@IRR**

berechnet die interne Ertragsrate für eine Reihe von Cashflow-Werten, die sich aus einer Investition ergeben. Im allgemeinen handelt es sich bei @IRR um die Raten, die dem aktuellen Wert einer erwarteten künftigen Folge von geraden oder ungeraden Cashflows für die ursprüngliche Investition entspricht. VIP geht davon aus, daß der Cashflow in regelmäßigen Abständen eingeht. Mein Beispiel sehen Sie in Abbildung 12.

Meine beiden Beispiele in dieser Tabelle zeigen zwei Arten der Anwendungsmöglichkeit von der @IRR-Funktion.

a.)@IRR(A2,B2..B14) gibt 6,11 % über eine 12-monatige Laufzeit zurück, wenn sich die erste Investition/ Zahlung auf DM 1000,— und die zwölf periodischen Eingänge auf jeweils DM 120,— belaufen. In Bereich B1..B13 enthält B1 den Wert 1000,— und B2..B13 enthalten jeweils den Wert 120,—.

b.)@IRR(C2,D2..D14) gibt 7,09 % über eine 12-monatige Laufzeit zurück, wenn sich die erste Investition/ Zahlung auf DM 1000,- und die zwölf monatlichen Einträge auf folgende Summen belaufen: 120, 124, 128, 132, 136, 130, 120, 124, 128 , 132 , 120, 124.

**@NPV**

berechnet den aktuellen Wert einer Reihe künftiger Cashflows. VIP geht davon aus, daß der periodische Zinssatz fest ist, und daß die Reihe von Cashflows (Zahlungsrückfluß) in gleichen Zeitintervallen stattfinden. Der erste Barabgang wird für das Ende der ersten Periode angenommen. Nachfolgende Cashflows finden am Ende der folgenden Perioden statt.

@NPV(ZINSSATZ,CASHFLOW-BEREICH)

Zinssatz ist der periodische Zinssatz. Cashflow-Bereich ist der Bereichsname oder die Adresse, die zur Speicherung der Serie mit Cashflows benutzt wird.

@NPV benutzt folgende Formel:

$$\sum \frac{V_i}{(1+\text{Zinssatz})^i}$$

wobei:

Vi .. Vn = Folge von Cashflows in Cashflow-Bereich
Zinssatz = Zinssatz
n = Anzahl von Cashflows
i = laufende Wiederholung (1 bis n)

Um den aktuellen Nettowert einer Investition zu ermitteln, bei der es sofort zu einem Barabgang kommt, auf den eine Reihe künftiger Bareingänge folgen, muß der ursprüngliche Abgang separat faktorisiert werden, da er von dem Zinssatz nicht betroffen ist.

Ist ANFANG der erste Barabgang, SERIE ein Bereich mit zukünftigen Cashflows und Zinssatz der periodische Zinssatz, so wird der gesamte aktuelle Nettowert folgendermaßen angegeben:

+ Anfang + @NPV (ZINSSATZ, SERIE)

**@PV**

die @PV-Funktion findet für die Berechnung des Barwertes einer Annuität Anwendung, dargestellt durch RATE, ZINSSATZ und ANZAHL der PERIODEN. Eine Annuität besteht aus einer Reihe von gleichen Raten, die in konstanten Zeitabständen gezahlt werden. Der Barwert ist der jetzige Wert aller Zahlungen, die später getätigt und um einen bestimmten Zinssatz diskontiert werden. Die Berechnung des Barwertes einer Annuität erlaubt Ihnen, verschiedene Investitionsmöglichkeiten oder

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Schätzung | Cashflow | Schätzung | Cashflow | |
| 2 | 0,05 | -1000 | 0.05 | -1000 | Ausgabe für Invest |
| 3 | | 120 | | 120 | Eingänge |
| 4 | | 120 | | 124 | " |
| 5 | | 120 | | 128 | " |
| 6 | | 120 | | 132 | " |
| 7 | | 120 | | 136 | " |
| 8 | | 120 | | 130 | " |
| 9 | | 120 | | 120 | " |
| 10 | | 120 | | 124 | " |
| 11 | | 120 | | 128 | " |
| 12 | | 120 | | 132 | " |
| 13 | | 120 | | 120 | " |
| 14 | | 120 | | 124 | " |
| 15 | | | | | |
| 16 | | | | | |
| 17 | | 6.11% | | 7.09% | |
| 18 | | \| | | \| | |
| 19 | | @IRR(A2,B2..B14) | | @IRR((C2,D2..D14) | |

*Abb. 12: Cashflow*

geht weiter...

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | RATE: | 1.500 | BARWERT: | 9,934,70 | |
| 2 | ZINSSATZ: | 14.00% | | \| | |
| 3 | PERIODEN: | 20 | | @PV(B1,B2,B3) | |
| 4 | | | | | |
| 5 | | | | | |

*Abb. 13: Barwert @PV*

potentielle Verpflichtungen hinsichtlich des Geldwertes miteinander zu vergleichen.

Das allgemeine Format der @PV-Funktion lautet:

@PV(RATE, ZINSSATZ, PERIODENZAHL)

Die allgemeine Formel zur Berechnung der Annuität heißt:

$$PV = Rate * \frac{1 - (1+Zinssatz)^{-n}}{Zinssatz}$$

In unserem Arbeitsblatt (Abbildung 13) wird die Formel folgendermaßen verwendet.

Der Unterschied zwischen der @NPV-Funktion und der @PV-Funktion beruht auf den Unterschieden bei den Zahlungsströmen und der Art, wie diese in der Tabelle angezeigt werden. @NPV berechnet den Barwert aus einer Zahlungsreihe, die gleiche und ungleiche Werte enthalten kann, die aber alle einem Zellenbereich in der Tabelle angehören. Die Raten in der @PV-Funktion müssen alle gleich sein, und dieser Betrag muß in einer einzigen Zelle stehen oder als Wert in die @PV-Formel eingegeben werden.

## @FV

diese Funktion entspricht in Ihrer Form der @PV-Funktion, wird aber zur Berechnung des Endwertes einer Annuität benutzt. Der Endwert ist der Wert einer Reihe von Ein- oder Auszahlungen zu einem zukünftigen Zeitpunkt, aufgezinst mit einem bestimmten Zinssatz. Die Berechnung des Endwertes einer Zahlungsreihe erlaubt Ihnen, verschiedene Investitionsalternativen oder potentielle Verpflichtungen miteinander zu vergleichen.

Das allgemeine Format der @FV-Funktion lautet:

@FV(RATE, ZINSSATZ, PERIODENANZAHL)

Die Formel zur Berechnung des Endwertes einer Annuität ist:

$$FV = \frac{(1+ZINSSATZ)^{-n}-1}{ZINSSATZ}$$

## @PMT

berechnet die periodische Zahlung eines Darlehens. Die meisten Abzahlungskredite werden wie Annuitäten einer nachschüssigen Rente berechnet, d.h. die Zahlungen werden am Ende jeder Zahlungsperiode vorgenommen.

*KAPITAL* entspricht dem Darlehensbetrag.

*ZINSSATZ* entspricht dem periodischen Zinssatz

*PERIODEN* entspricht der Anzahl der Zahlungsperioden

@PMT benutzt folgende Formel zur Berechnung der periodischen Zahlungen:

$$KAPITAL * \frac{ZINSSATZ}{1-(ZINSSATZ+1)}$$

wobei:

*KAPITAL* = Kapital
*ZINSSATZ* = Zinssatz
*n* = Perioden

Nehmen wir einmal an, bei Ihnen ist der Geldregen ausgebrochen und Sie entschließen sich eine Eigentumswohnung für 250.000,-DM zu kaufen. Die Finanzierung splittet sich in 2 Hypotheken zu DM 150000,— und DM 100000,—. Die Laufzeit beträgt im 1. Fall 30 Jahre, im 2.Fall 25 Jahre.

Für die Hypotheken erstellen wir folgende Formel:

@PMT(150000,6.75%/12,30*12) = DM 972.89

@PMT(100000,6.75%/12,25*12) = DM 690.91

Gesamtbelastung somit DM 972,89+DM 690.91 = DM 1663.80 pro Monat.

## Mach mal Pause!

"Kann ich nie, lerne ich auch nicht, kann ich nicht gebrauchen !" Ich höre es ja, Sie brauchen nicht auf mich loszuwettern. Es ist eben sehr umfangreich, dieses VIP-Professional, stellen Sie sich mal vor, es gibt auch noch den Teil Datenbank und Grafik, der ebenfalls behandelt werden möchte. Sie haben richtig gehört, unsere bis jetzt behandelten Formeln, Macros und Funkionen betreffen nur den Kalkulationsteil. Au weh ! Hoffentlich springen mir nicht die Leser ab, wenn ich Sie mit dem gesamten Sprachschatz konfrontiere.

## Walk on the wild Side

würde LOU REED sagen, wenn ich ihm wieder eine Hausaufgabe geben müßte. Aber ich werde Sie diesmal verschonen. Bitte schauen Sie sich nachfolgende Grafik an. Wie man so etwas macht, lernen Sie im nächsten Teil.

*Bei Problemen:*
*Heinz D. Schultz*
*Ingersheimer Strasse 12*
*7140 Ludwigsburg*

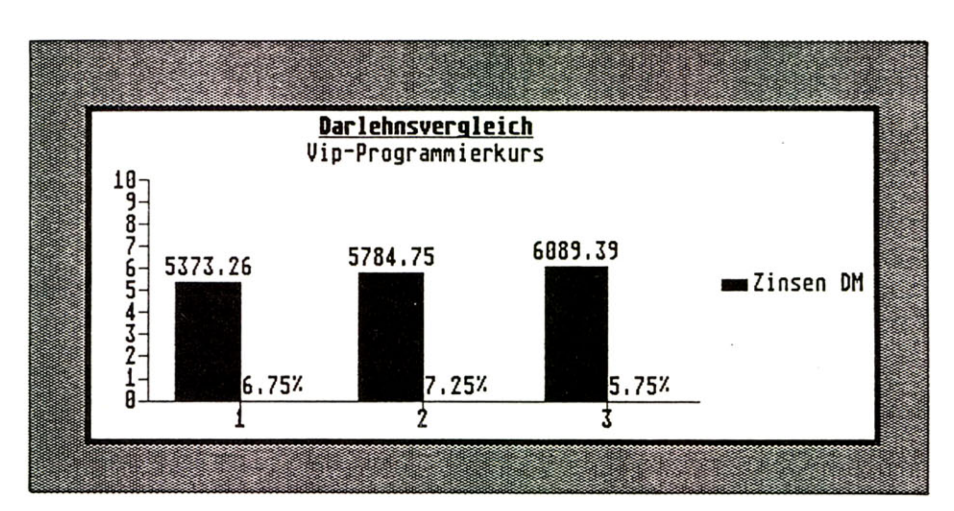



*Abb. 14: Grafik*

*geht weiter...*

***Lösungslisting zu unserer letzten Hausaufgabe ST6/88***

```
Zelle: Inhalt:
A1:    'Angebot
B1:    '|
C1:    1
D1:    2
E1:    3
A2:    'Darlehen in DM
B2:    '|
C2:    50000
D2:    50000
E2:    50000
A3:    \-
B3:    '|
C3:    \-
D3:    \-
E3:    \-
A4:    'Zinsen % p.A.
B4:    '|
C4:    (P2) 0.0675
D4:    (P2) 0.0725
E4:    (P2) 0.0575
A5:    'Gebuehr  1mal
B5:    '|
C5:    680
D5:    480
E5:    850
A6:    'Laufzeit MON.
B6:    '|
C6:    36
D6:    36
E6:    48
A7:    \-
B7:    '|
C7:    \-
D7:    \-
E7:    \-
A8:    'RATEN p.M.
B8:    '|
C8:    (F2) @PMT(C2,C4/12,C6)
D8:    (F2) @PMT(D2,D4/12,D6)
E8:    (F2) @PMT(E2,E4/12,E6)
A9:    \-
B9:    '|
C9:    \-
D9:    \-
E9:    \-
A10:   'Kreditsumme
B10:   '|
C10:   (F2) +C8*C6
D10:   (F2) +D8*D6
E10:   (F2) +E8*E6
A11:   'Zinsen Summe
B11:   '|
C11:   (F2) +C10-C2
D11:   (F2) +D10-D2
E11:   (F2) +E10-E2
```

**ENDE**

# VOM PAPIER ZUR DATENBANK

**Das Erstellen von Programmen In den vergangenen drei Kursteilen wurden Datenbankbefehle im interaktiven Modus eingegeben. Heute folgt die Programmierung. Umfangreiche Befehlsketten können abgespeichert und per Programm ausgeführt werden. Die beiden Datenbanksysteme erlauben das Verfassen überaus anspruchsvoller und bedienungsfreundlicher Endbenutzeranwendungen.**

```
SET TALK OFF
mtaste = ' '
DO WHILE mtaste = ' '
  CLEAR        (bei dBMAN bitte ERASE schreiben)
  @ 10,12 SAY 'Dies ist ein Test'
  @ 12,14 SAY 'Bitte drücken Sie eine Taste'
  @ 12,43 GET mtaste PICTURE '!'
  READ
  IF mtaste = 'A'
    @ 14,12 SAY 'Dies war Taste A'
  ELSE
    @ 14,12 SAY 'Dies war nicht Taste A'
  ENDIF
  DO CASE
    CASE mtaste = 'B'
      @ 16,12 SAY 'Es war Taste B'
    CASE mtaste = 'C'
      @ 16,12 SAY 'Es war Taste C'
    OTHERWISE
      @ 16,12 SAY 'Es war Taste '+mtaste
  ENDCASE
ENDDO
@ 20,12 SAY 'Damit wird die Schleifenbedingung logisch falsch'
WAIT
RETURN
```

*Bild 1: Übungsprogramm*

## Besonderheiten der Programmiersprachenkomponente

Die "eingebaute" Programmiersprache ist sehr leistungsfähig und kann mit anderen (BASIC, Pascal) sehr wohl konkurrieren bzw. übertrifft sie in Bezug auf die Verwaltung von Daten. Im Gegensatz zu diesen Programmiersprachen müssen die Programme hier aber streng strukturiert sein. Was bedeutet das?
Erstens: Es sind keine Sprunganweisungen möglich. Da die Programme ohne Zeilennummern (z.B. GFA-BASIC) geschrieben werden und das Benutzen von Labels nicht vorgesehen ist, kann innerhalb eines einzelnen Programms nicht hin- und hergesprungen werden. Der vom BASIC her bekannte "Spaghetti-Code" wird völlig vermieden. Sprünge sind lediglich in bzw. aus Unterprogramme(n) möglich. Eine unbedingte Verzweigung auf bestimmte Programmzeilen ist nicht möglich. Mit der Anwendung der Unterprogrammtechnik entstehen hierarchische Abhängigkeiten zwischen den einzelnen Modulen.
Zweitens: Programme bzw. Unterprogramme dürfen nur einen Ein- und Ausgang haben, wobei der Start immer die erste Zeile ist und das Ende die letzte Zeile des Programms sein muß. Dies ist keine Forderung der Programmiersprache sondern eine Regel der strukturierten Programmierung.
Drittens: Die komplette Steuerung von Wiederholungsstrukturen erfolgt ausschließlich mit kopfgesteuerten Schleifen, d.h. die Abbruchbedingung befindet sich am Anfang.
Weiterhin ist wie in GFA-BASIC nur eine Anweisung pro Zeile erlaubt.
Diese Einschränkungen erscheinen zunächst als Hindernis. Gerade das Verbot von Sprüngen bedeutet für BASIC-Umsteiger eine große Umgewöhnung. Mich selbst plagte einst das Problem, ein vorhandenes BASIC-Programm in dBASE III umzustricken (ich habe es dennoch geschafft !). Deshalb will ich auch die Vorteile nennen:

a) Übersichtliche und einfache Programmstrukturen
b) Reduzierung der Dokumentation (selbstsprechende Programme)
c) schnelle Änderungsmöglichkeiten
d) leichtere Einarbeitung fremder Programmierer
e) Reduzierung von Fehlern
f) weniger Aufwand beim Testen

## How To Do?

Nun soll es aber endlich losgehen! Um von dem interativen Modus in den Editor (jawohl, der ist eingebaut) zu gelangen, schreibt man in die Kommandozeile

*MODIFY COMMAND programm*

(programm) ist ein von Ihnen frei wählbarer Name der Programmdatei (denkt an TOS). In ihr wird der Code auf der Disk abgespeichert; die Extension des Files lautet automatisch .cmd. Das Programm wird in ASCII-Notation gesichert und kann bei Bedarf durch Anklicken im Desktop gelistet oder gedruckt werden. (Anmerkung: die Endung der Indexdatenbanken bei ST-BASE lautet .nmx und nicht .ndx.).
Nach diesem Aufruf befindet Sie sich in einem Full-Screen-Editor und das Tippen kann beginnen. Eingegebene Befehle

werden jetzt nicht sofort ausgeführt, sondern erst nach Aufruf der Kommando-Datei. Als Übungsbeispiel nehmen Sie bitte das Programm aus Bild 1. Es sind Anweisungen enthalten, die ich weiter unten beschreiben will. Machen Sie sich so bitte mit den besonderen Eigenschaften Ihres Editors vertraut.
Mit CTRL-S kommt man aus dem Ediermodus (dBMAN) zurück (bei ST-BASE: Pull-Down-Menü oder ESC-Taste und RETURN) zum Kommando-Interpreter. Der eben geschriebene Quellcode wird automatisch auf die Disk gesichert. Starten Sie dann das Programm und sehen Sie mit erstaunten Augen, was passiert. Dazu müssen Sie natürlich den Namen des Programms angeben.

*DO programm*

DO ist in etwa mit RUN (BASIC) vergleichbar. Bei BASIC ist das Programm bereits im Speicher; in unserem Beispiel muß der Quellcode noch von der Diskette geladen werden. Falls Änderungen erforderlich werden (Tippfehler seid gegrüßt), müssen Sie nochmals mit "MODIFY COMMAND ..." die ursprüngliche Datei laden und die Korrekturen vornehmen. Der fehlerhafte Text wird erneut geladen und steht zur Edierung bereit.

## Ausgewählte Anweisungen

Das Übungsbeispiel ist mit einer Fülle neuer Befehle geladen. Natürlich können auch alle bereits bekannten datenbankspezifischen Befehle (LIST, APPEND ...) auch innerhalb eines Programms mitbenutzt werden, die ich in diesem Fall weggelassen habe.
Nun zur Beschreibung des Programms im einzelnen.
Die einzig mögliche Schleife ist schnell erklärt. "DO WHILE bedingung" wiederholt die folgenden Befehle (bis zum ENDDO), solange die Bedingung wahr ist. Es wird der Logikzustand der Bedingung bei Einstieg in die Schleife abgefragt. Trifft der Interpreter auf "ENDDO", wird erneut die Bedingung untersucht und bei Wahrheit nochmals ausgeführt. Endlosschleifen kann man also einfach mit "DO WHILE .T." einleiten, weil das "T" für TRUE steht (siehe 3.Teil). Diese Schleifenbedingung ist so immer gültig. Es muß nicht eine Vergleichsbedingung (mit Gleichheitszeichen) sein wie in meinem Beispiel. Es kann auch der Logikzustand von logischen Variablen (z.B. DO WHILE .NOT. EOF()) zur Steuerung benutzt werden. Verändert man in der Schleife nicht die Bedingung, entsteht ein "Loop", aus dem man nur mit der ESCAPE-Taste herauskommt. Vorsicht! SET ESCAPE OFF schaltet die Funktion dieser Taste aus. Deshalb darf der Schalter erst dann aus sein, wenn das Programm fehlerfrei ist.
Sie sollten sich angewöhnen, alle Befehle, die zwischen DO WHILE und ENDDO stehen, einzurücken. So erkennt man schneller das Ausmaß der Wiederholungsstruktur. Zu jedem "DO WHILE ..." muß ein passendes "ENDDO" vorhanden sein, sonst spielt das Programm verrückt. Der Vollständigkeit halber weise ich auf den EXIT-Befehl hin, der den sofortigen Ausstieg aus der Schleife bewirkt, ohne die Bedingung zu prüfen. Der Programmablauf geht dann nach dem zugehörigen ENDDO weiter. Da dieser Befehl die mühevoll gepflegte klare Struktur eines Programms zerstört, bitte ich darum, EXIT nicht in Einsatz zu bringen. Durch geschickte Programmierung ist der Befehl sowieso überflüssig zu machen. Ich verstehe nicht, weshalb er in den Sprachschatz mit aufgenommen wurde.
CLEAR bzw. ERASE löscht den Bildschirm. Die SAY-Anweisung gibt formatierten Text an der entsprechenden Position (oben links ist Koordinate 0,0) aus. Die beiden Zahlen geben die Zeile und Spalte an. So können der Inhalt von Variablen und Dateifeldern gezeigt oder Überschriften erzeugt werden. Die Zahlen müssen nicht absolut sein. Die relative Adressierung ist auch zugelassen (z.B. zeile = 10, spalte = 5, @ zeile,spalte SAY ...). Von dem bekannten "?" (print) möchte ich abraten, weil mit ihm keine formatierte Ausgabe möglich ist. Die Umleitung der Ausgabe übernimmt der interne Schalter SET DEVICE TO PRINTER. Damit können formatierte Listings erstellt werden, denn die Koordinaten beziehen sich beim Drucker auf absolute Druckpositionen. Ohne großen Aufwand ist so das Bedrucken von z.B. Endlosüberweisungsträgern möglich. Standardmäßige Einheit (Device) ist der Bildschirm (SCREEN).
Mit GET ... READ können Eingaben von der Tastatur aus erfolgen. "GET" bereitet die folgende Variable auf die Eingabe vor, die auf jeden Fall in ihrem Typ schon definiert sein sollte. Das anschließende "READ" liest die Daten dann in die Variable ein. Die wahlfreie Erweiterung "PICTURE" gibt dabei das Eingabeformat vor, z.B. wandelt das Ausrufezeichen die Eingaben automatisch in Großbuchstaben um; Zahlen werden gar nicht erst zugelassen. Weitere PICTURE-Angaben:

z.B. *PICTURE '999.99'*
fünfstellige Zahl mit 2 Nachkommastellen

*PICTURE 'AAAAAA'*
sechs Alphazeichen, keine Zahlen

*PICTURE 'XXXXXX'*
alphanum. Eingabe, 6 Stellen

*PICTURE 'AAA-AA 9999'*
Auto-Nummernschild.

Für jedes einzugebende Zeichen muß bei der Verwendung von PICTURE ein Formatzeichen vorhanden sein! Bei mehreren gleichzeitig "offenen" GETs beendet ein einziger READ die Eingabe.
Damit ist also ohne großen Aufwand der Aufbau von Plausibilitätskontrollen möglich. Schon bei der Eingabe von Daten kann verhindert werden, daß z.B. in ein numerisches Feld Alphazeichen eingegeben werden (Resource Contruction Set ade!). Die verwendete Variable (hier: mtaste) kann auch ein Feld einer Datei sein, um die Daten gleich in die Datenbank zu schreiben. Es wäre alternativ denkbar, die Eingabe zunächst in eine interne Variable zu speichern, und dann (z.B. nach einer Prüfung oder Umrechnung) mittels des REPLACE in das Feld eines Satzes zu befördern.
Es stehen weitere Befehle für Eingaben zur Verfügung. Mit ACCEPT können auch Daten von der Tastatur übernommen werden. Ich selbst benutze diese Anweisung jedoch nie, weil die PICTURE-Option hierbei nicht angewandt werden kann.
In meinem Übungsbeispiel folgt nach der Tastaturabfrage (sie wird in der Variablen "mtaste" gespeichert) eine bedingte Verzweigung. Ist der logische Zustand der Bedingung nach "IF" wahr, wird die nächste Zeile bis zum ELSE ausgeführt. Bei Unwahrheit werden die Zeilen zwischen "ELSE" und "ENDIF" durchlaufen. Um die Transparenz des Programmlistings zu erhalten, sind diese Anweisungen ebenfalls um zwei Leerstellen eingerückt. Sie müsen doch zugeben, daß die Lesbarkeit mit dieser simplen Technik erheblich gesteigert wird. Der optionale "ELSE"-Zweig kann weggelassen werden.
Sind mehrere Abfragen nötig, kann der IF-Befehl leicht überfordert und unübersichtlich werden. Deshalb ermöglicht die "DO CASE"-Anweisung eine Mehrfach-Fallunterscheidung. Die entsprechenden

Zeilen nach einem "CASE" werden dann ausgeführt, wenn die vorherige Bedingung ein wahres Ergebnis liefert. Trifft keiner der Fälle ein, so führt der Interpreter die Befehlsfolge nach "OTHERWISE" aus. Ist dieses "Andernfalls" nicht erforderlich, kann die Zeile entfallen. Abgeschlossen wird die "DO CASE"-Anweisung mit ENDCASE.

Ich meine, für die (hoffentlich) englischsprechenden Programmierer dürften diese Befehle nicht zu schwer sein. Der Quellcode liest sich einfacher als englischsprachige Literatur. Die Sprachelemente sind von derartiger Transparenz, daß eine weitere Beschreibung nicht nötig ist. Man kann das Programm problemlos ins Deutsche übersetzen (viel Spaß).

Die drei Anweisungen (Wiederholung, Alternative, mehrfache Alternative) sind die einzigen Möglichkeiten, die Steuerung innerhalb eines Programms zu beeinflussen. Weitere Sprachelemente sind nicht vorhanden bzw. nötig.

Mir persönlich fehlt ein wenig die fußgesteuerte Schleife, in der die Abbruchbedingung am Schleifenende steht (wie in PASCAL, REPEAT...UNTIL...).

Zur symbolhaften Darstellung des Programmablaufs bietet sich die Notation nach Nassi-Shneiderman an. In der DIN 66261 sind die Sinnbilder aufgezeigt; der altbekannte PAP(Programm-Ablauf-Plan) ist für strukturierte Programme weniger geeignet. Ich will hier keine Einführung in das Zeichnen von Struktogrammen geben, sondern nur darauf hinweisen, wie optimal sich diese Darstellungsart mit der Programmiersprache ergänzt. Zur Verdeutlichung habe ich das Struktogramm zu dem Übungsprogramm gezeichnet.

## Unterprogramme

Eine weitere Forderung an strukturierte Programme ist, daß ein einzelnes Programm nicht zu lang (= Verlust der Übersichtlichkeit) sein sollte. Als Faustregel gilt: maximal zwei Schreibmaschinenseiten pro Programm. Kurze Module sind nicht nur leichter zu korrigieren und zu testen; das Blättern im Editor ist bei langen Quellcodes ein ernstzunehmendes Hindernis.

Mit Hilfe der Unterprogrammtechnik können umfangreiche Programme verkürzt werden. Ständig wiederkehrende Teile werden in ein Unterprogramm ausgelagert und bei Bedarf aufgerufen. Bild 2 zeigt den Einsatz dieser Technik. Das Einstiegsmenü der Anwendung

```
* Programm: MENU.CMD
mtaste = ' '
SELECT 1    (bzw. SELECT FJ)
USE datei INDEX index1
DO WHILE mtaste <> '0'
  CLEAR     (bzw. ERASE)
  @ 5,30 SAY 'Hauptmenü'
  @ 10,30 SAY '1 - Erfassen'
  @ 12,30 SAY '2 - Ändern'
  @ 14,30 SAY '3 - Anzeigen'
  @ 16,30 SAY '4 - Löschen'
  @ 19,30 SAY '0 - Programm-Ende'
  @ 22,30 SAY 'Bitte Ihre Wahl -->'
  @ 22,50 GET mtaste PICTURE '9'
  READ
  DO CASE
    CASE mtaste = '1'
      DO erfass
    CASE mtaste = '2'
      DO aender
    CASE maaste = '3'
      DO anzeig
    CASE mtaste = '4'
      DO loesch
  ENDCASE
ENDDO
RETURN

* Unterprogramm ERFASS.CMD
CLEAR (bzw. ERASE)
@ 5,30 SAY 'Erfassen neuer Daten'
APPEND BLANK
@ 10,30 SAY 'Name   ' GET fname PICTURE 'XXXXXXXXXXXXXXXXXXXX'
@ 11,30 SAY 'PLZ    ' GET fplz  PICTURE '9999'
@ 12,30 SAY 'Ort    ' GET fort  PICTURE 'XXXXXXXXXXXXXXXXXXXX'
@ 13,30 SAY 'Datum  ' GET fdatum PICTURE '99.99.99'
@ 14,30 SAY 'Größe  ' GET fgroesse PICTURE '99.99'
READ
REPLACE falter WITH YEAR(DATE())-YEAR(fdatum)
RETURN


* Unterprogramm ANZEIG.CMD
CLEAR (bzw. ERASE)
mname = '                    '
@ 5,30 SAY 'Anzeigen von Daten'
@ 20,30 SAY 'Wer soll gezeigt werden ?'
@ 21,30 GET mname PICTURE 'XXXXXXXXXXXXXXXXXXXX'
READ
GO TOP
SEEK mname
IF .NOT. EOF()
  @ 10,30 SAY 'Name   '+fname
  @ 11,30 SAY 'PLZ    '+STR(fplz,4)
  @ 12,30 SAY 'Ort    '+fort
  @ 13,30 SAY 'Alter  '+STR(falter,2)
  @ 14,30 SAY 'Datum  '+dtoc(fdatum)
  @ 15,30 SAY 'Größe  '+STR(fgroesse,4,2)
ELSE
  @ 10,30 SAY 'Daten konnten nicht gefunden werden'
ENDIF
WAIT
RETURN
```

***Bild 2: Grundsätzlicher Aufbau von Anwendungsprogrammen (Die einzelnen Module sind seperat abzuspeichern!)***

(MENU.CMD) ruft je nach Wunsch Module auf, die die verschiedensten Funktionen übernehmen.

Das Beispiel zeigt gut die entstehende Hierarchie. Oben befindet sich das Einstiegsprogramm; die Unterprogramme sind jeweils eine Stufe tiefer und vom Hauptprogramm abhängig. Ein Unterprogramm wird wie bekannt aufgerufen mit

*DO uprog*

Es muß natürlich bereits auf der Diskette vorhanden sein (UPROG.CMD); es wird nämlich nachgeladen. Nach seiner Abarbeitung wird es mit RETURN verlassen, und die Steuerung des Ablaufes geht in dem aufrufenden Programm in der nächsten Zeile weiter.

Es ist wichtig, auch das Hauptprogramm mit RETURN abzuschließen, um bei

```
I------------------------------------------------------------------I
I                                                                  I
I   I------------------------------------------------------I   I
I   I         Systemmeldungen ausschalten                   I   I
I   I------------------------------------------------------I   I
I   I         Leerzeichen -> mtaste                         I   I
I   I------------------------------------------------------I   I
I   I solange mtaste = ' '                                  I   I
I   I   I--------------------------------------------------I   I
I   I   I lösche Bildschirm                                 I   I
I   I   I--------------------------------------------------I   I
I   I   I Eingabe:-> mtaste                                 I   I
I   I   I--------------------------------------------------I   I
I   I   I                  mtaste = 'A' ?                   I   I
I   I   I-------ja------------I-----------nein-------------I   I
I   I   I                     I                             I   I
I   I   I "Dies war Taste A"  I  "Dies war nicht Taste A"   I   I
I   I   I                     I                             I   I
I   I   I---------------------I----------------------------I   I
I   I   I                  mtaste = ?                       I   I
I   I   I-------B------I------C------I------sonst----------I   I
I   I   I "Taste B"    I  "Taste C"  I  "Taste"+mtaste      I   I
I   I---I--------------I-------------I---------------------I   I
I   I "Damit wird die Schleifenbedingung logisch falsch"   I   I
I   I------------------------------------------------------I   I
I                                                                  I
I------------------------------------------------------------------I
```

*Struktogramm*

Programmende wiedersauber in den Kommando-Modus zu gelangen. Dieser Befehl verläßt immer eine niedrigere Hierarchie und kehrt eine Stufe höher zurück. Bitte gewöhnt Sie sich an, in jedem Programm nur EINMAL RETURN zu benutzen und das am Ende!! So ist sofort das logische und physikalische Ende eines Moduls zu erkennen.

## Das Grundprinzip aller Anwendungen

In Bild 2 habe ich ein "Gerippe" aufgezeigt. So in etwa sehen alle Anwendungen aus, die mit Hilfe eines Datenbanksystems geschrieben werden. Die vier Hauptfunktionen "Erfassen, Anzeigen, Löschen und Ändern" benötigt man in fast allen datenverarbeitenden Programmen (z.B. Plattenverwaltung, Adreßdatei). Für Eure spezielle Anwendung ist es nur noch vonnöten, dieses Gerippe mit entsprechendem "Fleisch" zu füllen. Die Funktionen "Ändern" und "Löschen" habe ich bewußt weggelassen; es steht Ihnen frei, sie selbst zu schreiben. Die Datei und Indextabelle stammen aus dem dritten Teil. Ich habe nach "meinen" Konventionen die Namen der Datenfelder mit einem vorangestellten "F" versehen. Übrigens, das Sternchen zu Beginn eines Unterprogramms ist eine Kommentarzeile; sie beeinflußt nicht den Programmablauf.

## Prozeduren

Gerade bei Diskettenbetrieb stört es ungemein, daß die Unterprogramme nach deren Aufruf erst von der Disk geladen werden, um zur Ausführung zu gelangen. Das Datenbanksystem hält nämlich immer nur ein Modul gleichzeitig im Speicher. Am Ende eines Unterprogramms sind wieder Ladeoperationen nötig.

Abhilfe schafft der Einsatz von Prozeduren. Sie werden dann eingesetzt, wenn das Programm komplett fertig geschrieben und getestet ist und für die Übergabe an den Anwender bereitsteht. Die Arbeitsschritte sind in etwa mit der Tätigkeit eines Linkers (=Klebers) zu vergleichen:

a) In jedes Modul (außer dem Hauptprogramm!) wird in die erste Zeile folgender Text geschrieben:

   *PROCEDURE name*

   (name) ist der Name der Kommandodatei, mit der sie aufgerufen wird, ohne die Endung .cmd (z.B. PROCEDURE ERFASS).

b) In das Hauptprogramm werden folgende Anweisungen am Anfang aufgenommen:

   *CLEAR ALL*
   *SET PROCEDURE TO haupt*
   *DO haupt*
   *CLOSE PROCEDURE*
   *RETURN*
   *PROCEDURE haupt*

   (haupt) ist der hier einzutragende Name des Hauptprogramms, mit welchem der Einstieg in die eigentliche Anwendung erfolgt.

c) Sind die Unterprogramme und das Hauptprogramm dergestalt modifiziert, müssen mit einem separaten Editor (z.B. Tempus) alle Teile zu einem großen File "zusammengeklebt" werden. Das Einstiegsprogramm muß ganz oben stehen. Diese Datei wird mit einem frei wählbaren Namen (z.B. procedur.cmd) auf Disk gesichert. Der Aufruf erfolgt dann vom Datenbanksystem wie gewohnt

   *DO procedur*

   Die komplette Prozedur wird geladen und im Speicher gehalten. Bei Aufrufen von Unterprogrammen treten keine Verzögerung mehr auf, weil sie nicht mehr nachgeladen werden müssen. Die Arbeitsgeschwindigkeit nimmt zu. Ein Prozedur-File kann nachträglich natürlich weiterhin wie jede andere Kommandodatei ediert werden, nur leider ist sie sehr umfangreich geworden.

## Das Ende naht!

Zum Abschluß die Auflösung der Aufgabe des dritten Teils. Es war gefragt nach der richtigen INDEX-Anweisung zum Sortieren eines Datumsfeldes, um eine Geburtstagsliste zu drucken. Sie lautet:

*INDEX ON STR(YEAR(datum),2) + STR(MONTH(datum),2) TO index4*

So wird die Datei aufsteigend nach Monaten und innerhalb eines Monats aufsteigend nach Tagen sortiert.

Damit wäre der Programmierkurs über relationale Datenbanken abgeschlossen. Ich hoffe, Ihnen das Handwerkszeug mitgegeben zu haben, um brauchbare Endbenutzeranwendungen zu schreiben (nicht nur für Sie selbst, sondern auch für Fremde). Mit der Zeit werden Sie sicherlich eigene Programmiertechniken entwickeln, um die Programme überaus komfortabel zu gestalten.

*Paul Fischer*


*Quelle:*
*DIN 66261:*
*Sinnbilder für Struktogramme nach Nassi-Shneiderman, November 1985*
*zu beziehen bei:*
*Beuth Verlag GmbH*
*Burggrafenstr.4-10*
*1000 Berlin 30*


ENDE



# ABO

ABO

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Bitte
mit
60 Pf.
frankieren

Postkarte

**Heim Verlag**
Heidelberger Landstr. 194

**6100 Darmstadt-Eberstadt**
Telefon 0 61 51 / 5 60 57

---



# Einzelheft- u. Monatsdisketten Bestellung



Einzelheft- u.
Disketten Service

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort



Bitte
mit
60 Pf.
frankieren

Postkarte

**Heim Verlag**
Heidelberger Landstr. 194

**6100 Darmstadt-Eberstadt**
Telefon 0 61 51 / 5 60 57

---



# Kleinanzeigen



Kleinanzeigen

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Bitte
mit
60 Pf.
frankieren

Postkarte

**Heim Verlag**
Heidelberger Landstr. 194

**6100 Darmstadt-Eberstadt**
Telefon 0 61 51 / 5 60 57



# Abonnement

Ja, bitte senden Sie mir die ATARI-Computer Fachzeitschrift ab ______________________
für mindestens 1 Jahr (11 Hefte) zum ermäßigten Preis von jährlich DM 70,– frei Haus.
(Ausland: Nur gegen Scheck-Voreinsendung DM 90,– Normalpost. DM 120,– Luftpost)
Der Bezugszeitraum verlängert sich nur dann um ein Jahr, wenn nicht 6 Wochen vor Ablauf des Abonnements gekündigt wird.

Name ______________________

Vorname ______________________

Straße/Nr. ______________________

PLZ ______ Ort ______________________

Gewünschte Zahlungsweise bitte ankreuzen

☐ Bequem und bargeldlos durch Bankeinzug

Konto-Nr. ______ BLZ ______

Institut ______ Ort ______

☐ Ein Verrechnungsscheck über DM______ liegt bei.

☐ Vorauskasse per Post-Einzahlung (Zahlkarte)

**Garantie:**
Diese Bestellung kann ich schriftlich innerhalb einer Woche (rechtzeitige Absendung genügt) widerrufen. Dies bestätige ich durch meine 2. Unterschrift.

Datum ______ Unterschrift ______

Datum ______ Unterschrift ______



## ABO



# Einzelheft- u. Monatsdisketten Bestellung

**ST-Computer können Sie direkt beim HEIM-VERLAG zum Einzelheft-Preis von DM 6,— (1986 + 1987) DM 7,— (1988) nachbestellen. Bearbeitung nur gegen beigefügten Scheck über den entsprechenden Betrag (keine Überweisung).**

| Jan. | Febr. | März | April | Mai | Juni | Juli/Aug. | Sept. | Okt. | Nov. | Dez. | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | 1986 = DM |
| | | | | | | | | | | | 1987 = DM |
| | | | | | | | | | | | 1988 = DM |

+ Gebühr für Porto u. Verp. = DM

(1 Heft DM 2,–, ab 2 Hefte DM 5,–)

☐ Scheck in Höhe zus. DM liegt bei

## Disketten Service

Alle Programme, die in ST-Computer veröffentlicht wurden, sind auf Disketten erhältlich. Die Disketten enthalten die Programme von jeweils 2 ST-Computer-Ausgaben. Bestellen Sie durch ankreuzen die gewünschten Disketten

| Preis je Diskette 28,– DM | Jan./Febr. | März/Apr. | Mai/Juni | Juli/Aug. | Sept./Okt. | Nov./Dez. | Jan./Febr. | März/Apr. | Mai/Juni |
|---|---|---|---|---|---|---|---|---|---|
| | 87 | 87 | 87 | 87 | 87 | 87 | 88 | 88 | 88 |

**Lieferung:** gegen beigefügten Scheck zuzügl. 5,— DM Versandkosten, unabhängig der bestellten Menge.



## Einzelheft- u. Monatsdisketten Bestellung



# Kleinanzeigen-Auftrag

**Bitte veröffentlichen Sie für mich folgende Kleinanzeige in der angekreuzten Rubrik**

**Biete an** ☐ Hardware ☐ Software  **Ich suche** ☐ Hardware ☐ Software ☐ **Tausch** ☐ **Kontakte** ☐ Verschiedenes ☐

30 Buchstaben je Standardzeile – incl. Satzzeichen und Wortzwischenräume.
Groß- und Kleinbuchstaben verwenden, fettgedruckte Wörter unterstreichen.

Bearbeitung **nur gegen Vorausscheck** über den entsprechenden Betrag (keine Überweisung)

☐ privat = DM 7,– je Zeile incl. MwSt.
☐ gewerblich = DM 15,– je Zeile + MwSt.
☐ Chiffregebühr = DM 10,–

☐ Scheck über DM______ ist beigefügt

Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze.

Datum ______ Unterschrift ______

Absenderangaben auf der Rückseite nicht vergessen

## Kleinanzeigen



# Kontaktkarte

## Kontaktkarte

Bitte Adresse der Firma, bei der Sie Informationen, oder etwas bestellen möchten, auf der rechten Seite eintragen → → → → → → → → → →

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ/Ort

Telefon Vorwahl/Rufnummer

**Postkarte**

Bitte freimachen

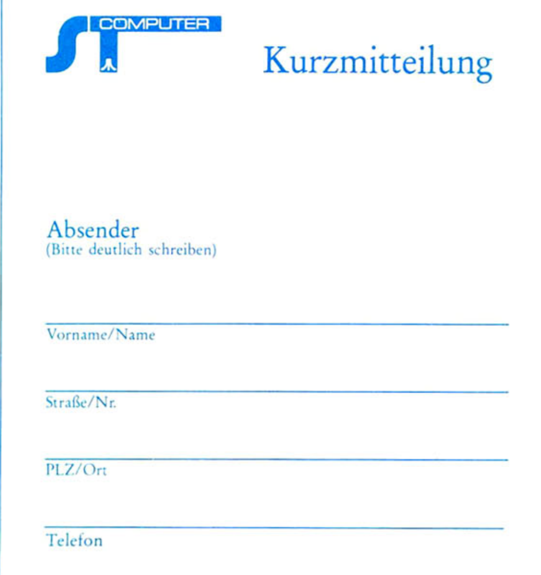
Firma

Straße/Postfach

PLZ Ort



# Kurzmitteilung



## Kurzmitteilung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Telefon

**Postkarte**

Bitte freimachen

MERLIN Computer GmbH
ST-Computer Redaktion
Industriestraße 26

6236 Eschborn



# PD Bestellung



## PD Bestellung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

**Postkarte**

Bitte freimachen

MERLIN Computer GmbH
ST-Computer Redaktion
Industriestraße 26

6236 Eschborn

ST COMPUTER

☐ Ich bitte um weitere Informationen

☐ Ich gebe folgende Bestellung auf

in Bezug auf Ihre Anzeige in ST-Computer Heft ______ Seite ___

| Menge | Produkt/Bestellnummer | DM | gesamt DM |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

Abgesandt am:

ST COMPUTER

Firma:

Bemerkungen:

---

ST COMPUTER

## Meine Meinung

Zu dem Artikel ______________ in Heft ____, Seite ____
hätte ich folgendes zu bemerken:

☐ Ich möchte Ihnen folgendes Programm zur Veröffentlichung anbieten: (Kurzbeschreibung, Sprache, Länge in Druckerseiten, GEM/TOS)

☐ Ich kann über folgendes Thema berichten: (Tips & Tricks am ST, Hardware, Software, etc.)

☐ Ich möchte gerne Autor in der ST-Computer werden. Meine Fachgebiete: (z.B. LISP, Pearl, Modula-2, Assembler ...)

☐ Ich möchte, daß folgendes Public-Domain Programm in Ihre Sammlung aufgenommen wird.

☐ Sonstiges

Bei weiteren Angaben oder Fragen wenden Sie sich bitte schriftlich oder telefonisch an die Redaktion. Tel. 0 61 96/48 18 11

ST COMPUTER

## Kurzmitteilung

---

ST COMPUTER

## PUBLIC DOMAIN SERVICE

Ich bestelle folgende PD-Disketten:
(Siehe PD Service in dieser Ausgabe)

Zahlung erfolgt:

☐ per Scheck

☐ per Nachnahme

Je Diskette fügen Sie bitte einen Betrag von DM 10,– bei,
für Porto und Verpackung je Sendung DM 5,– (Ausland DM 10,–)

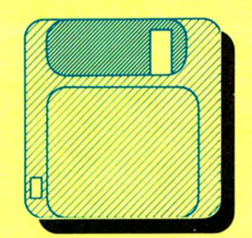
Datum

Unterschrift

ST COMPUTER

## PD Bestellung

Datum



# 120 NADELN MACHEN DRUCK

## Fünf DIN A3-Drucker im Test

**Was haben der Kontorist, die Sekretärin und der Chef gemeinsam? Einen DIN A3 breiten Drucker an ihrem Rechner. Ob Lagerlisten erstellt, Briefe und Umschläge gemeinsam gedruckt oder Umsatzstatistiken verlangt werden; oft reicht ein Drucker der normalen Breite (DIN A4) nicht mehr aus. Fünf solcher Drucker stellten sich unserem Test.**

Vorweg: Der Anwendungsbereich für die breiten Drucker liegt nicht nur im Büro. Haben Sie sich nicht auch schon oft genug darüber geärgert, daß der Ausdruck Ihrer Listings reichlich zerfetzt aussah, weil die Zeilenlänge nicht reichte? Hatten Sie nicht schon einmal den Wunsch, eine Graphik in A3 auszugeben? Auch sind die Preise für solche Geräte durchaus nicht nur am Investitionsvermögen einer Firma orientiert. Wem A3 dennoch zu breit ist, dem sei gesagt, daß es die meisten der Geräte auch in schmaleren Versionen gibt.
Doch kommen wir zu den Prüflingen. Für den Test standen uns zur Verfügung: Ein Brother M-1724L, Citizen HQP-45, C.Itoh C-715A, OKI Microline 393C und ein Seikosha SL-130 AI. Die Preise der Geräte sind genauso unterschiedlich wie Ihre Ausstattung. Sie reichen von 1899 DM für den Seikosha bis zu 3298 DM für den C.Itoh. Angesichts solcher Differenzen stellt sich natürlich die Frage: Welche Anforderungen stellt man an einen Drucker? Welche Kriterien muß ein Drucker, ungeachtet seines Preises und seiner Ausstattung, erfüllen? Lesen Sie dazu den Kasten über Testverfahren. Eine komplette Aufstellung über Ausstattung und Preise der fünf Geräte finden in der entsprechenden Tabelle.



*Das Testfeld beim Sonnenbad*

Die getesten Drucker lassen sich, was Preise und Leistung betrifft, klar in zwei Gruppen teilen: Zum einen die ca. 2000 DM teuren Brother, Citizen und Seikosha. Der C.Itoh und der OKI kosten zwar rund 1000 DM mehr, warten aber, wie wir noch sehen werden, mit teilweise besonderen Features auf. Wie man in der Tabelle sieht, sind einige Ausstattungsmerkmale, die noch vor einigen Jahren nur absoluten Spitzenmodellen vorbehalten waren, heute in der Preisklasse über 2000 DM schon fast Standard. Dazu gehören wechselbare Fontmodule, durch die man nicht mehr an den eingebauten Zeichensatz des Druckers gebunden ist, sondern ihn gegen einen nachzukaufenden austauschen kann. Da fällt der Seikosha auf, der bereits ab Werk über 9 Fonts verfügt. Für ihn bietet Seikosha keine weiteren Fonts an, was bei einer solchen Ausstattung aber auch nicht nötig ist. Bei dem für den Brother erhältlichen Modul handelt es sich um eine ganze Platine, die nachgerüstet wird. Neben drei neuen Zeichensätzen erwirbt der Kunde mit ihr auch gleich noch eine 30 kByte-Speichererweiterung.
Ebenso modular handhaben einige Anbieter die Kombatibilität ihrer Produkte.

Hieß es früher: 'emuliert einen X-Drucker', so lauten heute die Verheißungen: 'kann je nach Kundenwunsch ein X-, Y- oder Z-Drucker emuliert werden'. Einige der Geräte bieten diese Möglichkeit, verschiedene Druckertypen zu simulieren, bereits ohne Zusatz, bei anderen müssen wieder entsprechende Karten oder dergleichen zusätzlich erworben werden. Im Kasten über Kompatibilität lesen Sie mehr zu diesem Thema.

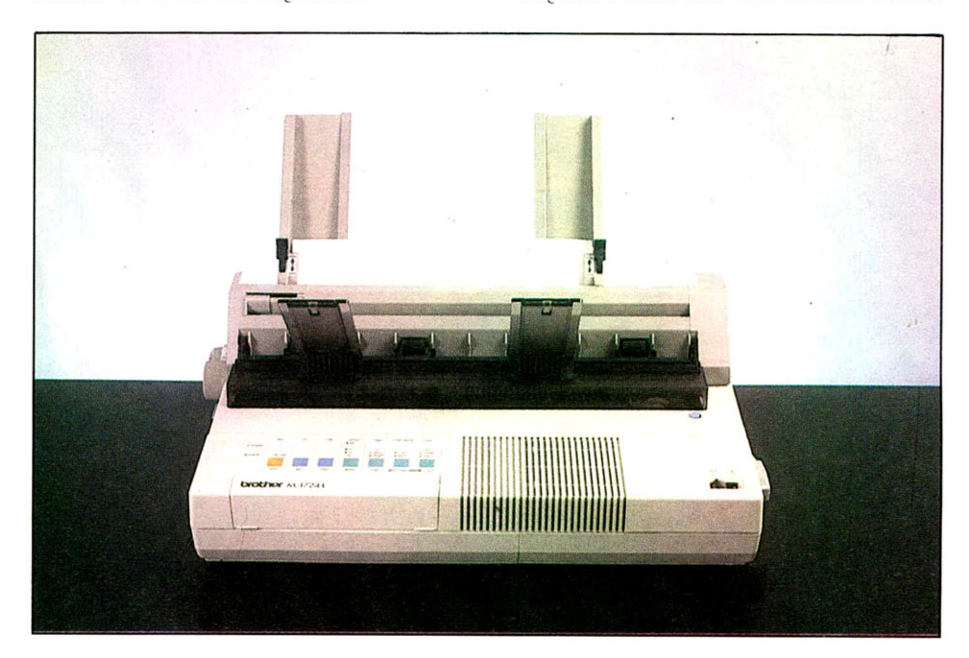
Papierpark-Funktionen bieten außer dem Citizen alle getesteten Geräte. Diese Funktion dient dazu, gleichzeitig Endlos- und Einzelblätter im Drucker zu bearbeiten. Beim Wechsel von Endlos auf Einzelblatt wird nur das Endlospapier zurückgezogen und der Traktor, der es antreibt, von der restlichen Mechanik entkuppelt.

## Die Tests

In dem Kasten über Testverfahren können Sie lesen, wie unser Druckertest-Programm Druckzeiten ermittelt. Da ist zum einen der Graphiktest, bei dem ein kleines Bildchen gedruckt wird. Der Test nach DIN 32751 wurde nach den Richtlinien der Norm durchgeführt. Den ca. 33 kBytes langen Test-Text haben wir dreimal gedruckt: einmal jeweils in Draft 12 CPI (Zeichen pro Zoll) und Letter Quality 10 CPI auf Endlospapier und - falls möglich - noch einmal in Letter Quality auf Einzelblatt, das dann aber mit dem automatischen Einzug zugeführt wurde. Dabei wurden jeweils zwei Zeiten ermittelt: Die zur vollständigen Datenabnahme und die Gesamtzeit zum Druck der Dokumente. Die Ergebnisse finden Sie in der Tabelle Druckzeiten. Deutlich erkennbar teilt sich das Feld auch hier in die bereits oben beschriebenen beiden Gruppen, zumindest was die Druckzeit bei Text anbelangt. Bei Graphik schwinden die Differenzen.

Wir haben den Test zusätzlich mit einigen anderen neuen und älteren Druckermodellen durchgeführt, um einen Vergleich zu ermöglichen. Die Zeit für den Graphikdruck des Star NL-10 ist jedoch nicht mit den anderen zu vergleichen, da für 9-Nadel-Drucker eine andere Graphik verwandt wird. Überhaupt sollen die Werte dieses Druckers in der Tabelle nur als Anhaltspunkte dienen, um zu zeigen, um wie vieles schneller auch vermeintlich 'langsame' 24-Nadler im Vergleich zu 9-Nadlern sind.

Wie im Kasten schon erwähnt, sind aber alle anderen Kriterien, die in die Bewertung eines Druckers mit eingehen müssen, nicht oder schlecht durch Messung zu erfassen. Daher haben wir mehrere Personen mit den fünf Prüflingen konfrontiert. Die Handhabung etc. wurde von ihnen beurteilt. Diese Personen waren zum Teil nicht oder wenig mit dem grundsätzlichen Umgang mit Druckern vertraut, was beimEinsatz eines solchen Geräts am Arbeitsplatz wohl immer der Fall sein wird. Was bei diesen Tests besonders angenehm oder unangenehm auffiel, lesen Sie im folgenden.




## Brother M-1724L

Der Brother fällt sofort durch sein extravagantes Äußeres auf: Bunte Tasten, viele Leuchten, der Transportknopf ist links angebracht. Er erinnert an französische Automobile, die auch nicht selten ungewohnte Detaillösungen mit ebensolchem Design verbinden. Sein Gehäuse ist ein Paradebeispiel an Kompaktheit. Obwohl er DIN A3-breites Papier verarbeiten kann, ist er doch kaum größer als ein DIN A4-Drucker. Diese Kompaktheit jedoch geht auf Kosten der Handhabung. Besonders das Einlegen von Endlospapier erfordert größte Konzentration und eine ruhige Hand. Die Konstrukteure haben den Benutzer aber davor bewahrt, diese Prozedur häufig durchführen zu müssen, denn der Brother besitzt eine sehr zuverlässige Papierpark-Funktion. Nur wehe dem, der häufig Aufkleber druckt und daher auch das Endlospapier tauschen muß.

Alle Funktionen des Druckers sind über das durchdachte Bedienfeld gut erreichbar. Beim Einziehen des Papiers bewegt der Drucker den Papierandruckhebel selbst, so daß kaum ein Fehler unterlaufen kann. Auch der Papierstärkehebel ist vorbildlich an der Außenseite angebracht, genau wie der Hauptschalter oben auf dem Gerät sitzt. Das Einziehen der Einzelblätter ist sowohl automatisch wie manuell sehr einfach. Der automatische Einzug wird elektrisch erkannt, nach seiner Installation muß keine Grundeinstellung des Gerätes geändert werden.

Die Geräusche jedoch, die der M-1724L beim Bewegen des Papiers erzeugt, sind alles andere als ein Genuß. Auch nach längerer Arbeit mit ihm kommt immer wieder die Befürchtung auf, das Gerät zerfiele sofort in sämtliche Bestandteile. Die DIP-Schalter liegen versteckt unter einer Klappe im Innern des Gerätes und sind schlecht zu erreichen. Der Druckbereich ist schlecht ablesbar, da er durch eine undurchsichtige Klappe abgedeckt wird. Dagegen ist das Zeilenlineal bei der Verwendung von Einzelblatt sehr gut zu erkennen. Eine Abreißkante existiert leider nicht. Mit der Epson-Kompatibilität verhält es sich, wie im Kasten beschrieben: Der IBM-Zeichensatz ist nur im IBM-Modus erreichbar und der wiederum läßt sich nur durch DIP-Schalter wählen. Leider ist auch das Handbuch ein schlechtes Beispiel. Es fällt durch Unübersichtlichkeit und fehlenden Index auf. Der Brother bietet die Möglichkeit, Dokumente per Tastenfeld zu kopieren. Da sich sein Speicher auf 54 kByte ausbauen läßt, genügt das für einige Seiten. Weitere Besonderheit: Das Gerät erlaubt auch die Verarbeitung von Postkarten. Dafür muß lediglich ein Hebel (wiederum an der Außenseite) umgelegt werden.

| | Brother M-1724L | Citizen HQP-45 | C.Itoh C-715A | Oki Microline 393C | Seikosha SL-130 AI |
|---|---|---|---|---|---|
| **Ausstattung** | | | | | |
| Schnittstellen | ser/par | ser/par | ser/par | ser/par | par |
| Speicher | 24 kB | 24 kB | 32 kB | 64 kB | 16 kB |
| Fonts eingebaut | Courier [1] | Courier [2] | Courier [2] | Courier [2] | 9 versch. [3] |
| Druckqualitäten | Dr./LQ | Dr./Korr./LQ | Dr./LQ | Dr./Korr./LQ/HS Dr. | Dr./LQ |
| Emulation | Epson LQ/IBM/Diablo | Epson LQ | von Modul abhängig | von Modul abhängig | Epson LQ/IBM |
| erhältl. Emulation | - | IBM/Diablo | Eps.LQ/IBM/Diablo | Epson LQ/IBM | Diablo |
| Papiertransport | Schub/Friktion | Schub/Friktion | Schub/Zug/Friktion | Schub/Zug/Friktion [4] | Schub/Friktion |
| Papierparkfunktion | ja | nein | ja | ja | ja |
| Zuführung Endlosp. | hinten | hinten | hinten/unten [5] | hinten/unten [5] | hinten |
| Anzahl Schächte | 1 | 1/2 | 1/2 [6] | 1/2 | 1/2 [6] |
| Max. Papierbreite | 420 mm | 406 mm | 393,7 mm | 406 mm | 406 mm |
| Zeilen / Einzelblatt | 64 | 64 | 62 | 64 | 61 |
| Farbdruck | nein | nein | ja | ja | nein |
| Handbuch | deutsch | deutsch | englisch | deutsch | deutsch |
| Einstellung Grundwerte | DIP-Schalter | DIP-Schalter | LCD-Display | gedrucktes Menü | gedrucktes Menü |
| **Preise** | | | | | |
| Drucker | 1995 DM | 2298 DM | 3298 DM | 3219,30 DM [7] | 1899 DM |
| Einzelblatteinzug | 454,86 DM | 780 DM | 598 DM | 590 DM | 499 DM |
| Fontkarten | 285 DM [1] | 148 DM | 98 DM | 118,42 DM | - |
| Farbband | 34,20 DM | 65 DM | 49 DM/Farbe: 55 DM | 58 DM/Farbe: 84 DM | 35 DM |
| Druckkopf | ca. 350 DM | ? | 798 DM | ? | 695,40 |
| Garantie | ½ Jahr | 2 Jahre | keine [8] | ½ Jahr | ½ Jahr |

[1] Für den M-1724L ist eine Steckkarte erhältlich, auf der sich drei weitere Fonts sowie eine 30 kB-Speichererweiterung befinden.
[2] Weitere Fonts sind als Steckkarten erhältlich.
[3] Courier, Prestige, Boldface, Gothic, Light Italic, Orator, OCR-A, OCR-B
[4] Zugtraktor als Option erhältlich
[5] Von unten nur bei Zugtraktor.
[6] Einzelblatteinzug auf zwei Schächte aufrüstbar.
[7] Hinzu kommen 153,51 DM für ein Emulationsmodul.
[8] Die Gewährleistung übernimmt lt. C.Itoh der Händler.

*Tab. 1: Ausstattung und Preise der Testkandidaten*

Dokumentlänge Text: 33396 Bytes
Dokumentlänge Graphik: 32643 Bytes
Genannt: Zeit zur Datenabnahme / Gesamtzeit für Druck

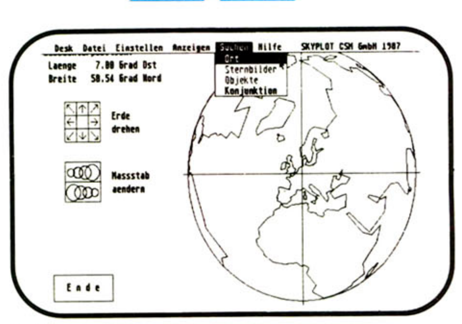
| | LQ-Einzel | LQ-Endlos | Draft-Endlos | Graphik | DIN 32751 |
|---|---|---|---|---|---|
| Brother M-1724L [1] | 0:18/12:02 | 0:18/10:47 | 0:23/4:47 | 0:16/0:21 | 38 s |
| C.Itoh C-715A | - | 0:18/8:31 | 0:19/3:41 | 0:18/1:02 [2] | 32 s |
| OKI 393C | 0:50/7:46 | 0:39/6:27 | 0:20/3:02 | 0:16/0:28 | 26 s |
| Citizen HQP-45 | - | 2:48/10:39 | 1:16/4:59 | 0:31/0:32 | 38 s |
| Seikosha SL-130AI | - | 5:21/10:08 | 2:42/5:06 | 0:17/0:25 | 36 s |
| Epson LQ-500 | 11:29/14:53 | 9:49/12:42 | 4:07/5:20 | 0:16/0:22 | 47 s |
| NEC P2200 | 14:14/18:41 | 10:37/13:50 | 5:56/7:40 | 0:27/0:34 | 51 s |
| Epson LQ-2500 | | 7:47/10:04 | 3:49/4:56 | 0:16/0:17 | 34 s |
| Star NL-10 | - | 25:13/29:59 | 6:29/7:35 | 1:32/1:33 [3] | 107 s |

[1] Der getestete M-1724L war mit einer 30 kB Speichererweiterung ausgestattet.
[2] Der C-715A druckt 180 DPI-Graphik in zwei Durchgängen pro Zeile.
[3] Vorsicht! Der NL-10 ist ein 9-Nadeldrucker!

*Tab. 2: Druckgeschwindigkeiten der Prüflinge und anderer Geräte.*

STARKE SOFTWARE

# Der absolute WAHNSINN:



Bestimmen des Standortes

SKYPLOT plus



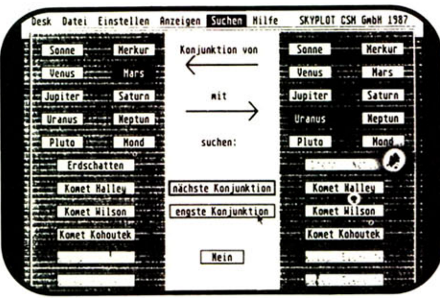
Was wird dargestellt (hier mit 15 383 Sternen)

## Technische Daten in Kürze:

- zwei Sterndatensätze mit 612 bzw. **15 383 Sternen**
- **1053 Nebel, Sternhaufen** und **Galaxien**
- Daten im ASCII-Format, veränder- u. erweiterbar
- acht Planeten, Sonne, Mond und drei Kometen vordefiniert
- Eingabe **eigener Bahnelemente** möglich, **elliptisch** und **parabolisch**
- Darstellung des Himmels als rechteckige Karte, Polarkarte, sichtbarer Himmel oder Horizontkarte in vier Himmelsrichtungen
- **Vergrößerungen** auf rechteckigen Karten mit der Maus bis **über eine Billion mal**
- **maßstäbliche Darstellung** von Nebeln und Planeten als Scheiben, letztere **mit Phasen** – etc. dadurch **Darstellung** von **Finsternissen, Durchgängen, Bedeckungen** etc.
- **automatische Suche** nach **Konjunktionen**, also Finsternissen, **Begegnungen, Bedeckungen** etc.
- **Bewegungsbahnen von Planeten** mit max. **32 000 Punkten**, speicherabhängig
- einstellbarer Datumsbereich **1.1.1583 – 31.12.3000**
- beliebige geographische Breite einstellbar
- Objektgruppen einzeln abschaltbar
- einstellbare Grenzgrößen für alle Arten von Objekten wirksam
- **Gradnetz** auf Wunsch **einblendbar**, Netzabstand einstellbar
- aktuelle Position des Mauszeigers auf der Karte kann angezeigt werden
- auf Farbmonitor Farben mausgesteuert veränderbar
- für **Hardcopy Bild invertierbar**

SKYPLOT plus nur DM 198,–

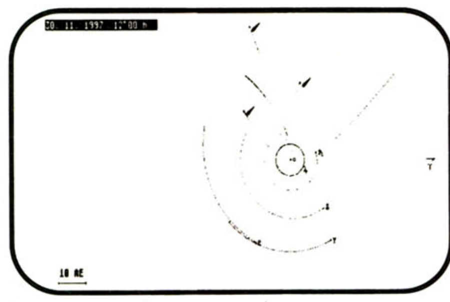


Konjunktion suchen

- **Darstellung dreidimensional oder von Planeten anderer Sterne aus**
- Berechnung der **Auf-** und **Untergangszeit** von Objekten, Kulmination etc.
- Darstellung des **Sonnensystems in vier verschiedenen Ansichten**, statisch oder mit **simultaner Bewegung** der Planeten
- **Stellarstatistik** zählt Sterne auf Ausschnitt und Gesamthimmel
- **Sichtbarkeitsdiagramme** für Helligkeit und Durchmesser der Planeten
- **Sternkarten** können **mit allen Objekten auf Plotter** ausgegeben werden
- Abspeichern einzelner Bilder oder Sequenzen für „PLANETARIUM"
- Löschen beliebiger Dateien zum Aufräumen von Disketten
- Simulation eines Filmes im Speicher, dadurch Bewegungsabläufe noch besser sichtbar
- eingebaute Kurzanleitung
- **Handbuchfunktion** verweist bei jedem Menüpunkt auf Handbuchseite
- **Alle ATARI ST mit mind. 1 MByte.**


Invertierter Himmel (hier 612 Sterne)



Sonnensystem in Bewegungssimulation

## So etwas gab's für den ATARI noch nie!

## BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

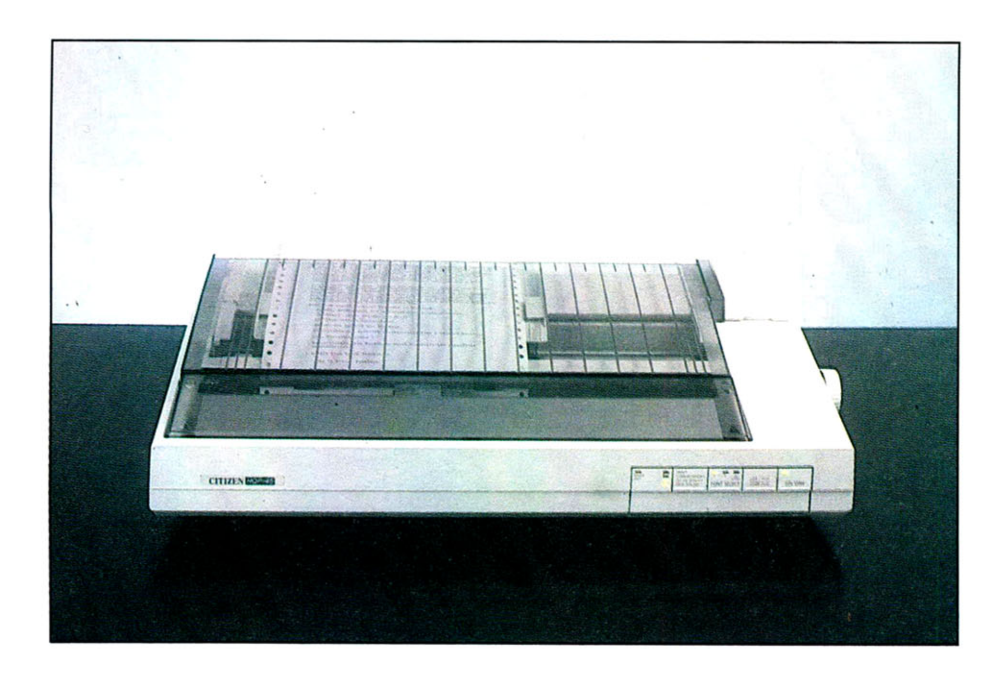
Ich bestelle _____ Stück SKYPLOTplus mit ausführlichem deutschen Handbuch à DM 198,– zzgl. DM 5,– Versandkosten (unabhängig von besteller Stückzahl) ☐ Scheck liegt bei ☐ per Nachnahme

Name, Vorname ______________________________

Straße, Hausnr. ______________________________

PLZ, Ort ______________________________

**Heim Verlag**
Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 0 6151-5 6057



## Citizen HQP-45

Der Citizen kommt im flachen, weißen Gehäuse daher, aus dem kein Hebel hervorragt. Sein Tastenfeld reduziert sich auf vier übersichtliche Tasten, über die leider nur die Schriftqualität und das Fontmodul gewählt werden können. Eine Taste für die Einstellung der Schriftbreite fehlt. Unterhalb des Tastenfeldes befindet sich an der Vorderseite die Klappe für die DIP-Schalter. Auch unter einer Klappe, jedoch an der Oberseite des Gerätes, finden sich die Schächte für zwei Fontkarten. Das Einlegen des Endlospapiers gestaltet sich als äußerst schwierig. Folgt man dabei den Anweisungen des Handbuchs, so verrenkt man sich fast die Finger, denn immer sind Teile der Einzelblattrutsche im Weg. Erst wenn diese entfernt ist, wird es etwas einfacher. Zu gut haben es die Konstrukteure mit einer Automatik zum Einzug des Papiers gemeint: Meldet der Papiersensor neues Papier, so beginnt der Drucker sofort, es weiterzuziehen. Das geschieht ungeachtet irgendwelcher Hebelstellungen, obwohl dafür der Papierandruckhebel erst in die 'Einrichten'-Position gebracht werden müßte. So wird manches Blatt in der Mechanik zerknüllt. Erschwerend kommt hinzu, daß just der entscheidende Papiersorten-Hebel unter einer Klappe liegt, die die gesamte Oberfläche des Druckers bedeckt. Das Einlegen von einzelnen Blättern gestaltet sich zwar erheblich einfacher, erfordert aber auch Fingerspitzengefühl. Hinzu kommt wieder die Automatik, die ohne Vorwarnung oder Befehl das Blatt einzieht. Der Citizen ist der einzige Drucker im Test, der über keine Papierparkfunktion verfügt, so daß die Mühsal des Einlegens von Endlospapier nach jedem Einzelblatt wieder ansteht.

Wie der Brother kennt auch der Citizen im Epson-Modus keinen Graphikzeichensatz. Hier muß dem ansonsten recht gelungenen Handbuch (Spiralbindung!) widersprochen werden, das den HQP-45 als 'absolut code-kompatibel zum Epson LQ-1000' bezeichnet.



## C.Itoh C-715A

Der C-715A ist der teuerste Drucker unter den fünf getesteten. Sein Preis von 3300 DM weist eher in Richtung Büro als beispielsweise die 1900 DM des Seikosha. Die erste Überraschung erlebt der C.Itoh-Besitzer beim Anheben des Druckers: 17 kg wollen bewegt werden und das bei einem recht kompakten Gehäuse. Ein erster Blick unter die Klappen bestätigt die Vermutung: Die komplette Mechanik besteht aus Metall, dieses Gerät scheint für den Dauereinsatz konzipiert. Die nächste Überraschung: Das zwar ausführliche Handuch ist leider vollständig in englischer Sprache verfaßt.

Vor dem ersten Probelauf muß dem Testobjekt aber eine Identität verpaßt werden: Wir wählen Epson LQ-1000 durch Einschieben der entsprechenden Karte. Nach problemlosen Einzug des Endlospapiers und dem Einschalten tut der Drucker seinen Status bereitwillig auf einem LCD-Display kund. Mit Hilfe dieses Anzeigefeldes werden die gesamten Grundeinstellungen der Maschine getätigt. Dabei fällt nur negativ auf, daß die Anzeige nicht beleuchtet ist und sich daher nur in einem bestimmten Winkel gut ablesen läßt.

Die Papierpark-Funktion des Testgerätes weigerte sich standhaft, ihre Bestimmung zu erfüllen. Wahrscheinlich war ein defekter Sensor daran schuld. Immerhin konnten wir uns damit behelfen, das Papier von Hand zurückzudrehen und den Traktor dann zu entkuppeln. Die Führung für die Einzelblätter steht in krassem Gegensatz zur Robustheit des Gerätes darunter: sie wirkt instabil und ist aus dünnem Kunststoff gefertigt. Trotzdem gab es beim Einzug keine nennenswerten Schwierigkeiten. Der Drucker bewegt den Papierandruckhebel automatisch, womit wie beim Brother eine Fehlerquelle ausgeschlossen ist.

## Epson, IBM - was heißt hier kompatibel?

Hand auf's Herz: Was erwarten Sie von einem Drucker, den Ihnen der Händler als Epson-kompatibel verkauft hat? Mit Sicherheit werden Sie sagen, kann dieser Drucker alles das, was ein Epson auch kann. Vielleicht noch ein wenig mehr.

## Denkste!

Wer sich heutzutage - im Zeitalter der Epromkarten, Steckmodule und Speicherriesen - einen Matrixdrucker zulegt, der laut Hersteller einen Epson LQ emulieren soll (Das bedeutet nur, daß er auf die gleichen Befehle genauso reagiert), der kann sich auf ein böses Erwachen vorbereiten.

Befragen wir das Handbuch eines Epson LQ. Dort steht unter ESC t n: 'Dieser Befehl wählt den durch die Codes 128-255 benutzten Zeichensatz...

n=0: Kursiv-Zeichensatz
n=1: erweiterter Epson-Graphik-Zeichensatz

Was hier so schön Graphik-Zeichensatz heißt, entpuppt sich bei genauerem Hinsehen als nichts anderes als der sog. IBM-Zeichensatz #2. Das ist der Teil des Standard-Zeichensatzes der IBM-Maschinen mit den ASCII-Werten 128-255. In ihm sind u.a. die Blockgraphikzeichen enthalten, mit denen auf PC-Schirmen häufig Eingabemasken etc. gestaltet werden. Was aber viel wesentlicher ist: Hier sind die internationalen Sonderzeichen, von den deutschen Umlauten bis zum dänischen durchgestrichenen O, festgelegt. Vergleicht man nun den Atari-Zeichensatz in diesem Bereich mit dem von IBM, so kann man einige Gemeinsamkeiten erkennen. Die Umlaute haben identische ASCII-Codes: ü=129, ä=132, Ä=142, ö=148, Ö=153, Ü=154.

Nun werden Sie sich vielleicht fragen, was die Kleinkrämerei soll. Schließlich lassen sich Umlaute auch anders drucken, und zwar mittels der Umschaltung der internationalen Zeichensätze. Die Antwort ist schnell gefunden. Da man bei wirklich Epson-kompatiblen Druckern den Graphik-Zeichensatz per DIP-Schalter vorwählen kann, können mit ihnen Texte mit deutschen Umlauten direkt und ohne eine Druckeranpassung gedruckt werden, z.B. mittels der Desktop-Funktion 'Drucke'. Nur das 'ß' haben uns die Atari-Ingenieure an eine Stelle gesetzt, an der im IBM-Satz das Peseta-Zeichen steht.

Was nun aber tun, wenn ein Drucker die Wahl des oberen Teils des Zeichensatzes innerhalb des Epson-Modus nicht zuläßt? Man wird nicht umhinkommen, jeden Text, der Umlaute enthält, mit einer Textverarbeitung wie Wordplus zu drucken. Der richtige Druckertreiber für Wordplus macht dann z.B. aus einem 'Ä' die Befehlsfolge: Umschalten auf deutschen Zeichensatz und Druck von '['. Tut er das nicht, erscheint auf dem Papier nichts.

Die Konsequenz dieser zweifelhaften Epson-Emulation oder -Kompatibilität ist, daß Druckertreiber für einen Epson LQ, die sich logischerweise auf den oben beschriebenen Befehl verlassen, mit vermeintlich Kompatiblen nicht funktionieren. Die griechischen Zeichen, die es sowohl bei IBM wie bei Atari gibt, sind gar nicht zu erreichen. Ihr Händler wird Ihnen sagen, daß Ihr neuer Drucker doch auch einen IBM Proprinter emulieren kann. Wenn Sie also diesen Zeichensatz wünschen, bitte, dann sollten Sie nur den entsprechenden DIP-Schalter umlegen oder ein 'Emulationsmodul' kaufen. Das wird Sie erfreuen, bis Sie daran gehen, sich für Ihren, jetzt IBM-kompatiblen Drucker neue Treiber für sämtliche Programme zu schreiben. Denn die Befehle, denen Ihr geliebtes Ausgabegerät eben noch gehorcht hat, mißversteht es nun. Auch die Suche nach einer Möglichkeit, zwischen den Modi per Software-Befehl umzuschalten, scheitert.

Es erweist sich so manche Werbeblättchen-Kompatibilität bei genauerem Hinsehen als eine Pusteblume, die im Wind zerfällt. Wer sich davor schützen möchte, der sollte vor dem Kauf das Handbuch des Auserwählten genau auf die Existenz des ESC t - Befehls untersuchen. Dieser muß im Epson-Modus die Zeichensatz-Umschaltung bewirken. Fehlt er, so ist es mit der Kompatibilität nicht weit her.

| | Brother M-1724L | Citizen HQP-45 | C.Itoh C-715A | Oki Microline 393C | Seikosha SL-130 AI |
|---|---|---|---|---|---|
| Handhabung:[1] | | | | | |
| Endlospapier | - | - | o | + | o |
| Einzelblatt | + | o | + | + | o |
| Tastenfeld | + | + | o | + | o |
| Geräusch | - | - | - | o | o |
| Sonstiges: | + Andruckhebel automatisch<br>+ Alle Hebel außen<br>- Druckbereich nicht lesbar | - keine Parkfunktion für Endlospapier | + Zugtraktor eingebaut<br>+ Andruckhebel automatisch<br>+ Robust | + Andruckhebel automatisch<br>+ Robust<br>- lauter Lüfter | + Hauptschalter gut zugänglich<br>+ Farbbandwechsel einfach<br>- Andruckhebel nur von innen |

[1] + = gut, o = durchschnittlich, - = schlecht

*Tab. 3: Testergebnisse auf einen Blick*

Das Geräusch, das der C.Itoh bei seiner Arbeit absondert, gefiel den Testenden genauso wenig wie die fehlende Abreißkante für das Endlospapier. Letzteres findet aber eine Begründung in der besonderen Konzeption des C-715A. Er birgt in seinem Innern einen Traktor, der seine Arbeitsweise mit zwei Handgriffen von Schub auf Zug verwandelt. Daher kann das Endlospapier auch durch den Boden des Druckers zugeführt werden (z.B. aus einem darunterstehenden Schrank). So mußten die Konstrukteure bei C.Itoh dafür sorgen, daß Papier in mehreren Richtungen durch den Drucker laufen kann. Das Bedienfeld ist ein Beispiel dafür, daß Aufwand sich nicht immer positiv auswirken muß. Das LCD-Display bietet aufgrund der schlecht bedienbaren Tasten und der mangelnden Übersichtlichkeit weniger Vorteile als beispielsweise das des Brother, der seine Einstellungen durch Lämpchen vermittelt. Auch beim C-715A fehlt die Möglichkeit, die Schriftbreite über Tasten zu wählen. Eine weitere Eigenart begegnet dem Anwender nach jedem kontrollierenden Blick ins Innere: Noch wenn die Klappe bereits geschlossen ist, meldet die Elektronik beharrlich: 'Cover Open'. Erst ein befreiender Druck auf die On-Line Taste schafft Abhilfe. Fehlt aber nun das Papier am Sensor, so wird folgerichtig 'Paper Empty' gemeldet.

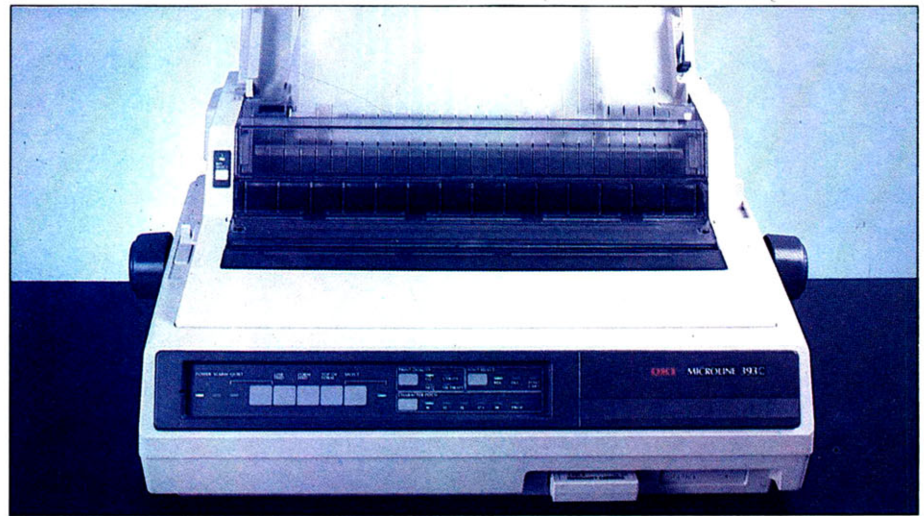
Verhieß die Aufschrift auf der 'Identity Card' noch Kompatibilität zum Epson LQ-1000, so stellte sich beim Test mit unserem Druckertestprogramm heraus, daß auch der C-715A nichts vom Graphikzeichensatz des Epson wissen wollte. Für den verglichen mit den anderen Kandidaten - hohen Preis bietet der C.Itoh noch die Möglichkeit, farbig zu drucken. Leider beschränkt sich mangels geeigneter Treiber am ST die Einsatzmöglichkeit dafür auf farbige Texte.

## OKI Microline 393C

Im Preis wie im Gewicht steht der OKI dem C.Itoh kaum nach. Das gilt genauso für die Ausstattung. Für ihn stand uns auch der einschächtige Einzelblatteinzug zur Verfügung, der -genau wie beim Brother - durch ein Kabel mit dem Drucker verbunden wird. Dadurch braucht sich der Benutzer um nichts mehr zu kümmern; die Maschine weiß nun, daß ein Einzug montiert ist. Überhaupt schien der OKI sehr 'schlau'. Wechselt man von Endlospapier auf Einzelblatt, so meldet er beileibe nicht 'Paper Out', sondern wartet geduldig auf den ersten Seitenvorschub, den er notfalls beim Druckbeginn selbst durchführt. Dabei bewegt er wie einige seiner Kollegen den Papierandruckhebel selbst. Ein sauber gegliedertes Anzeigefeld informiert (zwar nicht üppig) über Schriftbreite, -qualität und Status des Geräts. Die Tasten sind leicht zu bedienen und - was besonders auffiel - es bedarf keines Umschaltens in den Off-Line-Status, um z.B. einen Zeilenvorschub auszulösen. Warum machen das eigentlich nicht alle Hersteller so? Eine solch 'mitdenkende' Elektronik erspart manche Verwirrung auf seiten des Benutzers.

Sämtliche Testpersonen gaben dem OKI in punkto Bedienung beste Noten, einzig der sehr tief liegende Druckbereich, der daher, wie auch das Zeilenlineal, schlecht lesbar ist, trübten das Bild. Mit dem OKI bereichert man die Sammlung der Schreibtisch-Püsteriche um einen weiteren, denn die kühlende Luft führt ihm ein nicht gerade leiser Lüfter zu. Darüber vermag auch die reichliche Schall-Isolation nicht hinwegzutäuschen. Negativ fiel auch auf, daß die Papierpark-Funktion sich nicht mit der automatischen Einzelblattzuführung verträgt. Sollte das einzig und allein daran liegen, daß dann keine Öffnung mehr vorhanden ist, durch die das Endlospapier nach außen transportiert werden kann?

Für den Microline 393C ist ein Zugtraktor auf Wunsch erhältlich. Ist er montiert, so kann - wie beim C.Itoh - das Papier auch von unten in den Drucker geführt werden. Das 'C' in seinem Namen weist auf die Möglichkeit des Farbdrucks hin. Diese kann aber am ST, wie gesagt, nur für Texte sinnvoll genutzt werden. Ansonsten zeigte sich der OKI mit montiertem Epson LQ-1000 Emulationsmodul als wirklich Epson kompatibel. Er beherrscht den Graphik-Zeichensatz (IBM 2) wie ein echter LQ und ist daher problemlos mit jeder für diesen Drucker gedachten Anpassung zu betreiben. Dieser Zeichensatz läßt sich auch per Menü dauerhaft aktivieren. Leider ist im Preis von 3120 DM kein solches Modul inbegriffen, es schlägt mit weiteren 153 DM zu Buche.



## Seikosha SL-130AI

Wie eingangs erwähnt, ist der Seikosha der preiswerteste Drucker in unserem DIN A3-Test. Trotzdem ist er, verglichen mit den übrigen Kandidaten, erstaunlich komplett ausgestattet. Vor allem die neun eingebauten Fonts machen ihn für Liebhaber der Schriftvielfalt interessant. Sein günstig (wie auch der Hauptschalter) auf der Oberseite gelegenes Tastenfeld hält den Benutzer über gewählten Font, Schriftbreite und Druckstatus auf dem Laufenden. Leider ist es etwas klein geraten, sodaß die Bedienung dadurch schwierig wird. Auch die Überlegenspause von ca. einer Sekunde, die der Drucker nach jedem Tastendruck einlegt, bedarf einiger Gewöhnung.

Ansonsten ist der Seikosha einfach zu bedienen. Die Papierparkfunktion versieht tadellos ihren Dienst. Der Druckbereich ist sehr gut - auch bei geschlossener Klappe -zu erkennen. Vorbildlich einfach ist der Wechsel des Farbbandes. Es handelt sich dabei um eine kleine Kassette, die mit dem Druckkopf hin und her bewegt wird. Die Geräuschentwicklung, die die Arbeit des Druckers begleitet, ist durchschnittlich; der Komfort wird aber erheblich dadurch geschmälert, daß der Papierandruckhebel nicht von außen bedient werden kann. Hier wurde schlicht und ergreifend ein Hebelchen weggespart, denn die Öffnung dafür ist bereits im Gehäuse vorgesehen. Daher muß vor jedem manuell zugeführten Einzelblatt die Abdeckung des Druckbe-

# Testverfahren oder: Wie schnell sind Drucker?

Jeder, der sich einmal mit dem Kauf eines Druckers getragen hat, weiß, daß die Hersteller die Geschwindigkeit ihrer Geräte in CPS (Zeichen pro Sekunde) angeben. Dieser Wert jedoch entbehrt jeden Bezug zur Wirklichkeit, gibt er doch nur die maximal erreichbare Geschwindigkeit innerhalb einer Zeile an. Verzögerungen durch Abbremsen des Druckkopfs und Zeilenvorschübe werden von ihm nicht erfaßt.

Diesen Mißstand haben auch die Hersteller erkannt und so ist es zu neuen Prüfverfahren gekommen. Aber - wie sollte es auch anders sein - bisher konnte man sich nicht auf ein einheitliches Verfahren einigen. Eine Gruppe japanischer Produzenten (IPS) präsentierte unlängst auf der CeBIT in Hannover ihren Standard. Nach ihren Vorstellungen sollen praxisähnliche Dokumente (Text, Tabelle und Graphik) gedruckt und die Geschwindigkeit jeweils in Sekunden pro Seite angegeben werden. Damit zogen sie sich jedoch den Unmut einer Gruppe europäischer Hersteller zu, die ihrerseits einen Standard vorschlagen. Geht es nach ihnen, so sollen zwar ähnliche Prüfdokumente gedruckt werden, die Geschwindigkeit jedoch in Seiten pro Stunde angegeben werden. Dieses Verfahren ist auch in den Vorschlag zur Neufassung der DIN 32751 eingegangen.

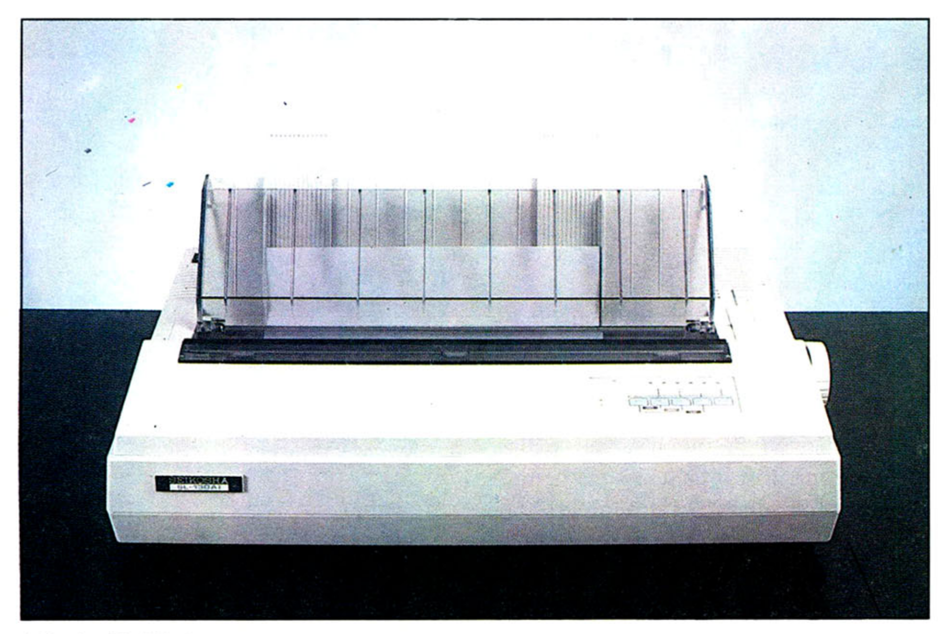
Die DIN 32751 schreibt in ihrer noch gültigen Fassung ein Testverfahren für EDV-Drucker vor. Der Test besteht aus dem Druck eines Normbriefes ('Dr. Grauert'-Geschäftsbrief nach DIN 5008). Neben der Zeit für den Druck ('Schreibzeit') sollte selbstverständlich noch die verwendete Druckqualität (Draft/Letter Quality) angegeben werden. Die Geschwindigkeit beim Druck von Graphik wird also momentan genausowenig ermittelt, wie die Verzögerung durch einen evtl. montierten Einzelblatteinzug beim Druck mehrerer Seiten erfaßt wird.

Bei all diesen Bemühungen um einen Standard beim Druckertest dürfen wir natürlich nicht vergessen, was der Kunde zu Haus mit seinem Drucker macht. Aus unseren Überlegungen dazu hat sich ein weiteres Prüfverfahren, unser eigener Standard, entwickelt.

## Ein Testprogramm wird geboren

Text und Graphik, so lautete die Devise, müssen wohl in einen Druckertest eingehen. Nur wie soll der Text aussehen, wie die Graphik? Wir haben uns für einen 33 kByte langen Text entschieden, die Anleitung zum Public-Domain-Editor 'Edimax'. Dieser Text hat viele Einrückungen und Zeilenvorschübe, die es dem Drucker nicht leicht machen, ihm aber auch die Möglichkeit zur Optimierung geben. Ähnlich die Graphik. Sie hat aus den genannten Gründen viel weiß. Wesentlich beim Druck sind zwei Zeiten: Zum einen die Gesamtzeit für den Druck selbst. Zum anderen aber nimmt der Drucker je nach Größe seines Speichers die zu druckenden Daten mehr oder weniger schnell ab. Nach dieser Zeit können Sie bereits mit dem Rechner weiterarbeiten, während der Drucker seinen Speicherinhalt ausdruckt. Beide Zeiten gehen in unseren Test ein. Was wäre eine Computerzeitschrift, die ein solches Verfahren nicht in ein Programm kleidet? Wir haben daher ein Druckertestprogramm erstellt, das es Ihnen ermöglicht, unser Testverfahren zu Hause nachzuvollziehen und Ihren eigenen Drucker genauso zu testen. Dieses Programm ist fortan im Public-Domain Service der ST-Computer erhältlich. Es beinhaltet außerdem noch das oben beschriebene Testverfahren nach DIN 32751 und eine Reihe von Tests, die das Erstellen eigener Druckeranpassungen erleichtern sollen.

Natürlich ist Geschwindigkeit nur eine Seite eines Druckers. Andere Merkmale, die genauso wichtig sind, wie Handlichkeit, Qualität des Handbuches, Lautstärke etc. sind jedoch nur bedingt oder gar nicht meßbar. Derlei Bewertungen können also nur durch komplizierte Meßverfahren (Frequenzanalyse des Druckgeräuschs) oder den subjektiven Eindruck mehrerer Personen gewonnen werden.



*Seikosha SL-130AI*

reichs geöffnet und der Hebel von Hand bewegt werden. Weitere Kritik gilt dem Handbuch. Besonders das Menü, mit dem die Grundeinstellungen des Geräts getätigt werden, wird nur kurz (und fehlerhaft) beschrieben. Erklärungen zu den einzelnen Punkten sucht man vergeblich. Nach unserem Testprogramm ist der SL-130AI absolut Epson-kompatibel, wie es sein 'Standard-Modus' auch verspricht. Leider kann der IBM-Zeichensatz 2 nur per Software aktiviert werden. Eine Einstellung im Menü, die ihn nach jedem Einschalten des Druckers zur Verfügung stellen würde, fehlt. So können zwar auch am ST Umlaute ohne Anpassung gedruckt werden, nur muß vorher der berüchtigte ESC t-Befehl an den Drucker gesandt werden; und das verlangt dann eben doch einen Treiber. Aber immerhin: Aufgrund der Existenz dieses Befehls arbeitet der Seikosha problemlos mit jedem Treiber für einen Epson LQ. Alle Freunde des PC-Ditto wird freuen, daß das Gerät bereits eine IBM-Emulation eingebaut hat.

## Der Wunsch-Drucker

Von jedem ein wenig, das wär' er. Die Handlichkeit vom OKI, das Tastenfeld vom Brother mit einer Anzeige wie beim C.Itoh... Wie immer, kann man auch bei Druckern nicht alles haben. So wird der zukünftige Kunde sich die Geräte, die ihm angeboten werden, sehr genau anschauen müssen. Wie unser Test zeigt, darf dabei die reine Druckgeschwindigkeit nur ein Aspekt sein. Dies zumal die noch übliche Angabe in Zeichen pro Sekunde mit Vorsicht zu genießen ist.
Vor allem muß man sich fragen, wozu der Drucker eingesetzt werden soll. Wie in der entsprechenden Tabelle zu sehen ist, sind die Druckzeiten für Graphik bei nahezu allen Geräten gleich, hier wird evtl. das ausgebende Programm aufgrund intensiver Rechenarbeit zu Verzögerungen beitragen. Bei Text jedoch bestehen teils sehr große Unterschiede. Aber auch hier ist zu fragen, ob der Text in Form von Listings, d.h. ohne besonderen Qualitätsanspruch gedruckt werden soll, oder ob große Mengen vorzeigbarer Dokumente bewältigt werden müssen. Für den Geschäftsbetrieb, für den unsere fünf Kandidaten ja eigentlich ausgelegt sein sollten, trifft wohl der letzte Punkt zu. Da muß vor allem Korrespondenz, am besten mit Durchschlag, erledigt werden. Die Menschen, die am Arbeitsplatz mit den Maschinen konfrontiert sind, haben zum einen nicht die Geduld des Freaks daheim, zum anderen werden sie auch nicht so bedacht mit dem ihnen zugeteilten Gerät umgehen. Für sie muß ein Drucker 'in der Hand liegen'. Der Wechsel von Papier muß einfach, die Bedienung insgesamt möglichst selbsterklärend sein. Außerdem sollte die Maschine ein gewisses Maß an Robustheit mitbringen, um nicht nach kurzer Zeit dem Büroalltag zu erliegen.

## Fazit

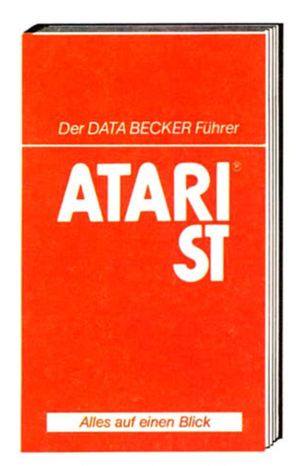
Unter den oben genannten Aspekten trennt sich unser Testfeld wiederum in die erwähnten zwei Gruppen. Betrachtet man Geschwindigkeit, Handlichkeit und Robustheit, so kommen für den professionellen Einsatz nur der OKI und der C.Itoh in Frage. Bei letzterem müssen aber Abstriche in der Bedienung gemacht werden. Der Brother hat alles Zeug dazu, auch in diese Gruppe vorzustoßen, doch seine mangelnde Druckgeschwindigkeit und die Labilität seines Gehäuses sprechen dagegen.
Der OKI wie der Brother zeigen, wie ergonomisch ein Gerät werden kann, wenn man nur will. Der erste Blick auf das Gerät sagt, wie es zu bedienen ist. Nach einmaliger Lektüre des Handbuchs versteht man die Papierpark-Funktion usw. Beim OKI fällt keine Klappe ab, wenn man sie öffnet. Bei anderen Geräten entsteht hingegen der Eindruck, daß die Entwickler mit ihren Produkten nicht ein einziges Mal gearbeitet haben. Da werden Geräte für runde 2000 DM angeboten, über die sich Ergonomen scheinbar weniger Gedanken gemacht haben als über einen Korkenzieher für 10 DM.
Die endgültige Entscheidung für oder wider ein Gerät hängt nicht zuletzt vom Geldbeutel ab. Ob Fontkarten und Emulationsmodule nun ein unbedingter Vorteil sind, darüber läßt sich streiten. Seikosha zeigt, daß verschiedene Schriftbilder - bereits eingebaut - nicht teuer sein müssen. Das Schriftbild selbst ist letztlich auch Geschmackssache und so liegt das letzte Wort dann doch wieder beim Kunden, der einen Kompromiß zwischen seinen Ansprüchen und seinen Finanzen schließen muß. Denn der Wunsch-Drucker - der muß erst noch gebaut werden.

*IB*

*ENDE*

# Mein Gott, jetzt hat er's...!

**Es passiert jedem einmal: Mitten in der Arbeit am ST fehlt eine bestimmte Information. Eine fürchterliche Sucherei beginnt. Irgendwann, meist Stunden später – der befreiende Schlag an die Stirn: „Jaaa, genau." Ersparen Sie sich doch diese unnötige Aufregung:**




Der DATA BECKER Führer zum ATARI ST – der ST-Anwender nennt ihn liebevoll „sein Mini-Intern". Doch das ist nicht weiter verwunderlich: In welcher Krise er sich mit seinem ST auch befinden mag, dieser DATA BECKER Führer hilft ihm weiter. Ob zur Hardware, zum Betriebssystem oder zu GEM – zuverlässig und schnell hat er hier noch jede Information gefunden. Immer auf einen Blick.
**Der DATA BECKER Führer zum ATARI ST**
**240 Seiten, DM 29,80**




Probleme mit den ST-Programmen? Ein Blick in den DATA BECKER Führer zu 1st Word, und schon kann die Arbeit weitergehen. Ganz gleich, ob Sie mit der englischen oder der deutschen Version arbeiten, hier finden Sie alle wichtigen Informationen. 1st Word, 1st Word Plus, 1st Lektor, 1st Mailmaster oder 1st Spooler – egal, worüber Sie mehr wissen wollen, hier finden Sie die Antwort.
**Der DATA BECKER Führer zu 1st Word**
**192 Seiten, DM 24,80**




Für alle GFA-Programmierer: Der DATA BECKER Führer zu GFA-BASIC. Schnell und gezielt läßt sich mit ihm jeder Syntax-Error überprüfen. Denn hier finden Sie übersichtlich geordnet und gut strukturiert die komplette Befehlsübersicht zum GFA-BASIC. Inklusive aller nötigen Parameter und Syntax-Erläuterungen – natürlich auch zu der neuen Version 3.0!
**Der DATA BECKER Führer zu GFA-BASIC**
**254 Seiten, DM 24,80**




Der DATA BECKER Führer zu SINUM! hilft weiter: Installation, Laden, Grafiken, Spaltensatz, Fußnoten, Tastenprogrammierung, Funktionstastenbelegung, Druckertreiber, Fonteditor – sollten Sie mal nicht weiterwissen, hier finden Sie die Lösungen. Natürlich auch mit Auflistung aller SIGNUM!-Fehlermeldungen.
**Der DATA BECKER Führer zu SIGNUM!**
**ca. 200 S.**
**DM 29,80**

**DATA BECKER**
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

EINSENDEN AN: DATA BECKER · MEROWINGERSTR. 30 · 4000 DÜSSELDORF 1
Hiermit bestelle ich für meinen ST
Name, Vorname
Straße
Ort
zzgl. DM 5,– Versandkosten unabhängig von der best. Stückzahl

# RELAX

## Hallo liebe Relaxer!

In diesem Monat gibt es wieder interessantes Spielefutter. Ich habe mir für Sie die Umsetzung des Automatenschlagers OUT RUN angesehen. Außerdem hat Relax noch zwei ausführliche Tests zu den beiden Superadventures MEVILO und Police Quest zu bieten. Natürlich gibt es in dieser Ausgabe auch wieder einen Wettbewerb. Diesmal verlost Activision 13 mal Knightmare. Zum Abschluß meines Vorworts möchte ich noch ein Gerücht dementieren, das zur Zeit vielen ST-Usern Kopfschmerzen bereitet. Einige Skeptiker behaupten, daß der ST in punkto Spielen eine tote Kiste ist. Man behauptet, es gäbe kaum noch Spiele für diesen Computer. Denkste! Jeden Monat habe ich aufs neue Schwierigkeiten, bei der Vielzahl der Games, die tagtäglich bei uns eintrudeln, die Spreu vom Weizen zu trennen. Alle englischen Softwarehäuser sowie alle französischen stellen munter ST-Games her. Alle Toptitel erscheinen früher oder später auch für Atari ST. Ein leidenschaftlicher Spielefreak kommt mit seinem Computer nicht zu kurz. Aber genug des Vorgeplänkels, nun geht es endlich los!

Viel Spaß
beim Lesen wünscht Ihnen

Ihr Carsten Borgmeier

# NEWS

Mit **Stock Market** von **Tynesoft** werden Sie in die Welt der Börsenspekulation eingeführt. In diesem gut gemachten Strategiespiel für Farbmonitore können Sie Aktien ankaufen, verkaufen, Kredite aufnehmen und vieles mehr tun, um ein erfolgreicher Spekulant zu werden.

*Info: Leisuresoft*

---

Mit **Stac (The ST Adventure Creator)** von **Incentive Software** kann man eigene Adventures edieren. Es läßt sich ein Parsersystem entwickeln, Texte für verschiedene Situationen schreiben und Grafiken entwerfen. Für Leute, die nicht mehr Adventures von professionellen Anbietern spielen möchten, sondern die Abenteuer lieber mit "Hausgemachten" erleben wollen, ist STAC gut geeignet. Eine ausführliche Dokomentation führt den Benutzer in das komplexe Programm ein.

*Info: Zum Redaktionsschluß kein deutscher Vertrieb bekannt.*

**Computer Hits** heißt eine Compilation von **Beau Jolly**, die für ST-User sehr interessant sein dürfte. Vier Spiele sind in der Verpackung enthalten. Darunter das dreidimensiomale Plattformspiel Brataccas, das Ballerspiel Deep Space, die gelungene Mischung aus Strategie- und Actionspiel Hacker II sowie der lustige Activisiontitel Little Computer People. In diesem originellen Programm kann man ein kleines Männchen beobachten, das in Ihrem ST wohnt, und mit ihm kommunizieren. Alle vier Titel waren schon auf dem internationalen Markt sehr erfolgreich. Eine Anschaffung der Spielesammlung lohnt sich in jedem Fall.

*Info: Ariolasoft*

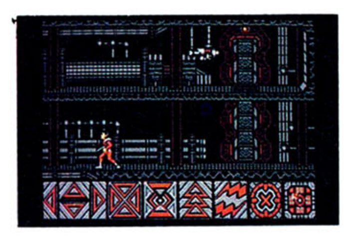
---




**Comicgrafik**, lustige Animationen und eine tolle Spielhandlung zeichnen Infogrames neustes Spiel aus. In **Bobo** muß man versuchen, als Knastbruder aus dem Bau auszubrechen. In sechs farbenfrohen Spielszenen kommt viel Freude auf.

*Info: Ariolasoft*

---



Der Käufer von **Obliterator** erhält neben Anleitung und zwei Disketten auch noch ein Poster als Zugabe. In diesem Spiel von **Psygnosis** steuert man Drak, einen unbesiegbaren Krieger, durch ein außerirdisches Kriegsschiff, um es zu vernichten. Währenddessen hat man sich mit allerlei tückischen Gefahren herumzuschlagen. Technische Fallen, wie Laserstrahltüren, aber auch unangenehme außerirdische Wesen, die das Feuer auf den Helden eröffnen, machen das Spiel schwer. Tolle Grafik und ein guter Sound sorgen dafür, das der Spieler nicht allzu schnell Maus und Joystick aus den Händen legt. Ariolasoft hat für alle englischmüden Computerspiele eine komplett ins Deutsche übersetzte Version zu bieten.

*Info: Ariolasoft*

---

**L'arche du Captain Blood** von **Infogrames** ist unter dem deutschen Titel die Arche des Captain Blood jetzt auch bei uns in Deutschland erhältlich.

*Info: Ariolasoft*

---

Blutig geht es bei **LEATHERNECK** von **MICRODEAL** zu. 1-4 Spieler können auf alle entgegenkommenden Soldaten ballern. Dazu stehen Handgranaten, Panzerfaust oder ordinäres Maschinengewehr zur Verfügung. Wenn ein Feind getroffen wird, ertönt ein schrecklicher Todesschrei. Die Bundesprüfstelle für jugendgefährdende Schriften in Bonn bekommt Arbeit.

*Info: Leisuresoft*

**Microdeal** hat sich etwas ganz besonderes für alle Goldrunnerfreaks ausgedacht. Zum kürzlich veröffentlichten Fortsetzungstitel **Goldrunner II** gibt es jetzt zwei Ergänzungsdisketten, auf denen sich Daten für neue Goldrunnerlandschaften befinden. Um sie verwenden zu können, benötigt man jedoch Goldrunner II.

*Info: Leisuresoft, Profisoft*

---

**Time Warp Productions** veröffentlicht bald ein Adventure mit dem Titel **Shadowworld**. Texteingaben sind nicht nötig. Per Maus werden alle Eingaben gemacht. Der Spieler steuert einen Reporter durch eine Abenteuerwelt und muß dabei viele Gefahren überstehen.

*Info:Rushware*

---

**Foundations Waste** heißt ein schlappes Ballerspiel, das zwar über einen guten Sound verfügt, grafisch aber etwas ärmlich aussieht. Das Raumschiff, das der Spieler bei vertikalem Scrolling steuern soll, ist zu klobig geraten.

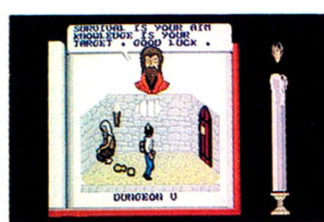
*Info:Profisoft*

---

**Captain America** von **Go** ist ebenfalls nicht sehr überzeugend. Als Spieler muß man versuchen, die Welt vor einer schrecklichen Katastrophe zu bewahren. Ein Verrückter namens Dr.Megalomann will den Präsidenten der Vereinigten Staaten erpressen. Wenn dieser kein Lösegeld bezahlt, will der Wahnsinnige die Welt in die Luft jagen. Keine Panic! Comicheld Captain America ist ja zur Stelle. Man steuert den Helden auf Plattformen und ballert dabei auf die Wächter des verrückten Megalomann. Damit das Spiel nicht gar so langweilig wird, hat der Hersteller der Verpackung eine Musikcassette beigelegt.

*Info:Profisoft*

---



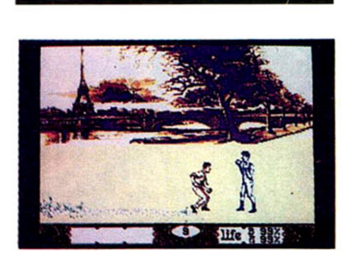

**Knightmare** heißt ein neues Game von **Activision**, das den Spieler in ein altes Verlies, tief unten in einem Schloß, versetzt. Mit Hilfe des Spielers soll er aus dem Gebäude fliehen. Es ist möglich, den Helden mit dem Joystick zu steuern und ihn per Tastatur einige Kommandos (wie beispielsweise Tür öffnen etc.) zu geben.

*Info: Leisuresoft, Ariolasoft*

**Crazy Cars** von **Titus** ist eine große Enttäuschung. Die akzeptable Grafik vermag nicht die spielerischen Mängel zu kaschieren. Das Autorennen ist tötlich langweilig. Mit einem Mercedes fegt man über einen amerikanischen Highway (Out Run läßt grüßen). Sie brauchen keine Angst um das Luxusgefährt zu haben. Sie können so viele andere Autos rammen, wie Sie wollen. Es passiert gar nichts!

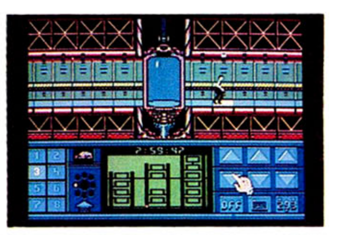
*Info: Leisuresoft*

---



In **Bob Winner** von **Loriciels** läuft man von links nach rechts über den Bildschirm und vermöbelt dabei alle Leute, die einem entgegenkommen. Ein langsames und deshalb auch langweiliges Spiel.

*Info: Leisuresoft*



**Impossible Mission II** von **U.S. Gold** ist ein Lichtblick unter den vielen Spielen, die im letzten Monat erschienen sind. Kenner des ersten Teils wissen Bescheid. Oberschurke Elvin will mal wieder die Welt zerstören. Als Spezialagent des Geheimdienstes müssen Sie das böse Unterfangen verhindern. Der Schurke hat sich in sein Hauptquartier zurückgezogen, das aus mehreren Türmen besteht. Man hüpft genau wie im ersten Teil mit Salti auf Plattformen in den verschiedenen Zimmern herum und untersucht alle Möbelstücke nach wertvollen Informationen. Dabei wird man von Elvins Robotern gestört. Der Spieler von Impossible Mission hat viele Aufgaben zu bewältigen. Zunächst muß er für jeden Turm einen dreistelligen Zahlencode finden, in dem er alle Möbelstücke untersucht. Dann muß man Tresore ausfindig machen und darin befindliche Musikstücke stehlen und diese zu einer Melodie zusammenfügen, mit der man die Tür zu Elvins Turmkommandozentrale öffnen kann. Es gibt viel zu tun, packen Sie es an!

*Info: Leisuresoft*

---

# Adventureperle Mevilo

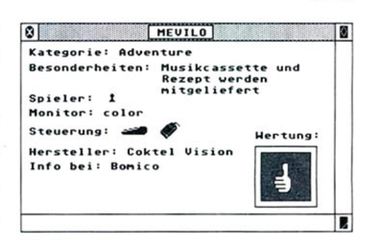



In MEVILO, dem neuesten Adventure des französischen Softwarehauses COKTEL VISION, spielt man die Rolle eines berühmten Parapsychologen in einer aufregenden Geschichte. Als Spezialist in Sachen Gespenster und Zombies bekommen Sie einen Auftrag von Ihrem Clienten Michel Hubert Destouches. Im Hause von Monsieur Destouches spukt es. Dumpfe Klopfgeräusche und unheimliche Seufzer versetzen den Armen in Angst und Schrecken. Ihr Client ist der festen Überzeugung, daß es sich beim Inititiator der unheimlichen Vorgänge um einen Zombie handelt. Also reisen Sie an den Ort des Grauens, auf die Insel Martinique, um dem Geheimnis auf die Spur zu kommen. Dabei soll geklärt werden, woher der Zombie kommt und was er will. Während des Spiels sind Sie damit beschäftigt, Gegenstände zu untersuchen, Personen auszufragen und die Insel Martinique zu erkunden. Da kommt Adventurefieber auf. Keine Angst, bei Mevilo brauchen Sie nicht stundenlang auf der Tastatur herumzuhämmern, bevor sich etwas tut. Einen Großteil der Aktionen werden per Maus gesteuert. Wenn man einen Gegenstand untersuchen möchte, bewegt man den Mauszeiger dorthin, und der Computer gibt in einem Fenster mit deutschem Text Auskunft, was es mit dem Gegenstand auf sich hat. Um den Ort zu wechseln, braucht man nicht "N", "S", "W" oder "E" einzugeben. Auf einer kleinen Karte am linken Bildschirmrand kann man mit Hilfe des Mauszeigers den gewünschten Ort aussuchen. Auf den Spieler warten eine Vielzahl von Überra-

schungen. Es gibt viele Rätsel zu lösen.

So befindet sich auf der Insel beispielsweise ein alter Mann, der sich perfekt in der Geschichte des Landes auskennt. Der Insulaner stellt Ihnen Fragen über die Insel, die Sie im Multiple-Choice Verfahren richtig beantworten müssen. Die Fragen sind eine harte Nuß. Wußten Sie beispielsweise, daß man Begasse zur Produktion von Zucker verwendet? Durch genaues Lesen der Anleitung und durch intensives Probieren kommt man aber den richtigen Antworten auf die Spur. Wenn man dem alten Insulaner alle Fragen richtig beantwortet hat, schenkt einem dieser eine Flasche. Was es damit auf sich hat, konnte ich während meines Tests nicht herausfinden. Nach dem Rätselstündchen beim alten Insulaner schaute ich mir eine Plantage an und besuchte meinen Auftraggeber Monsieur Destouches. Dort sollte ich mit meinen parasychologischen Fähigkeiten zu einer verstorbenen Frau Kontakt aufnehmen. Mevilo präsentiert sich in einer außergewöhnlichen, farbenfrohen Grafik. Soundeffekte, wie das digitalisierte Grunzen eines Schweins oder französische Musik sorgen für eine gelungene Atmosphäre. Coktel Visions neuestes Adventure versteht es, den Spieler für lange Zeit zu unterhalten. Mevilo ist eine Bereicherung für den Softwaremarkt. Durch das Spiel kann man viel über Martinique erfahren. Man lernt einen Teil der französischen Kultur kennen, muß Indizien kombinieren und hat mit der spannenden Handlung viel Spaß. Kein Wunder, daß Mevilo schon eine hohe Auszeichnung bekommen hat. Der französische Premierminister Chirac verlieh der Autorin von Mevilo zur Veröffentlichung die silberne Medaille der Stadt Paris. Als Bonus liegt der Verpackung eine Cassette mit französischer Musik und ein Rezept für ein französisches Spezialgericht bei. Liebhaber von anspruchsvollen Computerspielen kommen mit Mevilo voll auf ihre Kosten.

*CBO*




# Out Run endlich für den ST

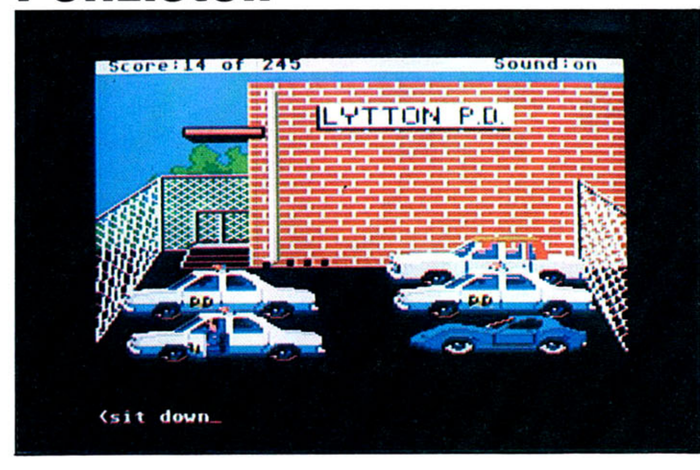



Wollten Sie nicht immer schon einmal einen Ferrari Testarossa fahren? In OUT RUN von U.S Gold haben Sie die Möglichkeit dazu. Mit einem hübschen Mädel an der Seite rast man unter Zeitdruck über einen amerikanischen Highway und überholt Käfer, Lastwagen und andere Verkehrsteilnehmer. Während der Bewältigung verschiedener Strecken erklingt eine packende Melodie. Sie stimmt den Spieler auf die Lust nach grenzenloser Freiheit ein. In der Spielhalle ist OUT RUN sehr erfolgreich. Nach den ersten weniger gelungenen Adaptionen des Automatenknüllers glaubte kaum einer daran, noch einmal eine vernünftige Heimcomputerversion dieses süchtig machenden Autorennens in die Hände zu bekommen. Amstrad und C64-Version waren im Vergleich zur Spielhallenfassung ein Totalausfall. Natürlich ist die die neue Atari ST Version dem Automaten ebenfalls nicht ebenbürtig. U.S. Golds Programmiererteam, Probe Software, ist jedoch mit der ST-Version eine ganz beachtliche Umsetzung gelungen. Die Grafik ist rasant und weis durch farbenfrohe und detailreiche Darstellungen zu überzeugen. Soundmäßig ist bei der ST-Version bis auf Kleinigkeiten ebenfalls alles in Ordnung. Satte Klänge klingen dem Spieler aus dem Lautsprecher des Monitors entgegen. Im großen und ganzen verfügt das ST-Out Run über dieselben Features, wie das Vorbild aus der Spielhalle. Lediglich das Radio zur Einstellung verschiedener Hintergrundmusiken fehlt. Dadurch wird dem Spielspaß aber kein Abbruch getan. Wem der Sound im Programm nicht reicht, kann ein neben Dokumentation und Spieldiskette mitgeliefertes Audiotape in seinen Cassettenrecorder einlegen und sich vom Original-Automatensound berieseln lassen.

*CBO*




# Aus dem Leben eines Polizisten



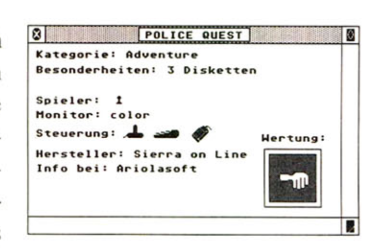

Das amerikanische Softwarehaus SIERRA ON LINE ist ein Spezialist in Sachen innovativer Adventuresoftware. Das zeigten Spiele wie KING'S Quest und LEISURE SUIT LARRY, aber auch das neueste Spiel von Sierra POLICE QUEST. In allen drei Adventures steuert der abenteuerlustige Spieler den Helden per Joystick durch eine komplexe Abenteuerwelt. Über Tastatur gibt man ihm Befehle, wie er in bestimmten Situationen zu reagieren hat. In POLICE QUEST übernimmt der Spieler die Rolle eines amerikanischen Polizisten namens Sonny Bond (Ähnlichkeiten mit Doppelnull James Bond oder Miami Vice Superbulle Sonny Crocket wären rein zufällig). Als Spieler begleitet man Sonny bei seiner tagtäglichen Arbeit. Das Geschehen beginnt in der Zentrale der Lyttoner Polizei. Sonny steht in einer Art Flur und wartet darauf, daß der Spieler seine Geschicke in die Hand nimmt. Punkt 13:00 muß man sich in einen Konferenzraum begeben, weil dort Sonnys Vorgesetzter über die kriminalistische Lage der Stadt berichtet. Es empfiehlt sich, sehr pünktlich zu erscheinen.

Bei der Polizei geht nämlich alle geordnet und diszipliniert zu. Wer da aus der Reihe tanzt, muß mit

einem abrupten Spielende rechnen. Auf einem Tisch im Konferenzraum liegt eine Zeitung. Diese ist nützlich, um Informationen über die Geschehnisse in der Stadt zu bekommen. Nach einigen Minuten Zeitunglesen marschieren die ersten Kollegen in den Raum und nehmen an den Tischen Platz. Wenig später tritt auch Sonnys Vorgesetzter ein. Alle Personen, denen Sonny im Spiel begegnet, sind animiert und haben einige Persönlichkeiten und somit eine bestimmte Bedeutung für den weiteren Spielverlauf. Die vielen Einrichtungen des Reviers wie Ermittlungscomputer, Funkanlage und Schreibmaschine werden zunächst nicht benötigt. Sonny fischt sich die Schlüssel vom Schlüsselboard und holt sich aus seinem Umkleideschrank Pistole, Notizblock, Munition und andere wichtige Gegenstände, die man als Streifenpolizist benötigt. Dann kann der Streifendienst beginnen. Man geht aus dem Gebäude zum Fuhrpark. Dort stehen einige Streifenwagen, aber auch Sonnys Privatwagen. Bevor man mit "drive car" einen Streifenwagen in Gang setzt, muß man zunächst einmal den Wagen überprüfen. Nachdem man um das Auto herumgelaufen ist und mehrmals "check car" eingegeben hat, setzt man sich ins Auto und fährt los. Der Computer fordert den Spieler auf, die zweite von drei Disketten in den Drive zu schieben. Hat man der Anweisung Folge geleistet, wechselt das Szenario. Man sieht nun die Stadt Lytton aus der Vogelperspektive. Einen kleinen Strich gilt es über ein Straßennetz zu steuern. Mit Hilfe von Funkcodes, die in der komplexen Anleitung aufgeführt sind, kann man in Kommunikation mit seinen Kollegen treten. Falls irgendwo ein Überfall stattfinden sollte, werden Sie durch geheime Funkcodes instruiert. Mit dem Joystick in der Hand steuert man über das Straßensystem, das sich über mehrere Bildschirme ausdehnt. Zur besseren Orientierung liegt der Verpackung ein Stadtplan bei. Die ganze Zeit mit einem Strich über einen Straßensystem zu huschen, wäre natürlich langweilig. Selbstverständlich besteht die Möglichkeit, den Wagen zu parken, diverse Gebäude zu betreten und sich dort ein bißchen umzusehen. Wenn man Zeuge einer Straftat geworden ist oder verdächtige Personen festgenommen hat, kann man sie verhaften und zur Wache bringen.

Police Quest bietet alles, was auch im Berufsleben eines Polizisten vorkommt. Da sind langweilige Streifentouren genauso inbegriffen wie aufregende Verfolgungsjagden, Verhöre und Ermittlungsarbeit mit dem Computer. Das Spiel zeichnet sich durch eine realistische Atmosphäre und einen guten Parser aus, der mit jeder sinnvollen Eingabe des Spielers etwas anzufangen weiß. Schwachpunkte liegen bei Grafik und Sound. In diesem Bereich bietet Police Quest nur Durchschnittliches. Die Handlung macht sehr viel Spaß. Sierras neuste Adventureschöpfung vermag den Spieler längerfristig zu unterhalten. Trotz Mängel bei Grafik und Sound bleibt Police Quest ein außergewöhnliches Spiel, das auch Adventureneulinge geeignet ist.

*CBO* *Infoadressen:*




## ADRESSEN

**Ariolasoft**
Postfach 1350
4830 Gütersloh 1
Tel: 05241/803871

**Leisuresoft**
Industriestraße 23
4709 Berkamen 5
Tel: 02389/6071

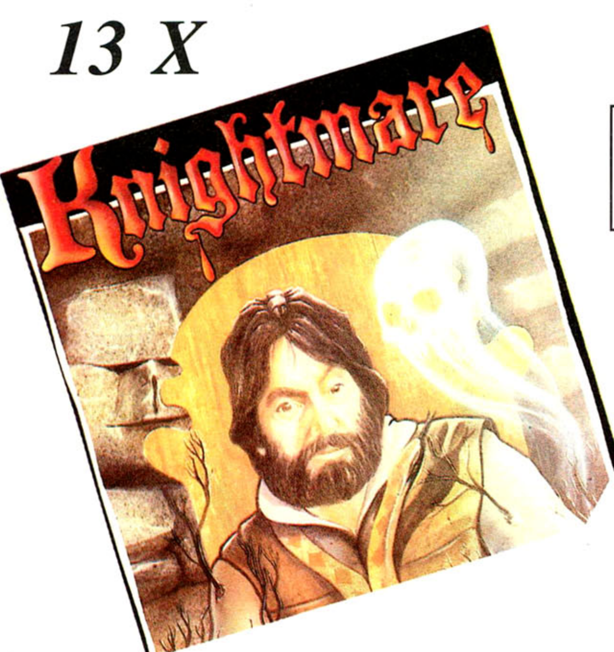
**Profisoft**
Sutthauser Str. 50/52
4500 Osnabrück
Tel:0541/53905

**Rushware**
Mircohandelsgesellschaft
Bruchweg 128-132
4044 Kaarst 2
Tel:02101/6070

**Bomico**
Elbinger Straße 3
6000 Frankfurt/M.90
Tel:069/706050

## Willkommen Ihr Wächter der Illusion, im Reich der totalen Konfusion...

# 13 X




# von ACTIVISION zu gewinnen!

Wie gewinne ich eines der 13 Knightmare von ACTIVISION?
Ganz einfach nur folgenden Satz richtig ergänzen!

Knightmare entstand aus der gleichnamigen faszinierenden ..................................................deren Basis traditionelle Abenteuer- und Phantasierollenspiele sind.

Die Antwort einfach auf eine Postkarte schreiben und an folgende Adresse schicken:

Merlin Computer GmbH
Stichwort: Knightmare
Industriestr. 26
6236 Eschborn

Einsendeschluß ist der 29.Juli 1988.

Mitarbeitern der Merlin Computer GmbH und des Heim Verlags ist die Teilnahme untersagt. Der Rechtsweg ist ausgeschlossen.

**KNIGHTMARE**

Knightmare ist eine neue faszinierende Fernsehserie der Anglia Television, deren Basis traditionelle Abenteuer- und Phantasierollenspiele sind. Sie werden mit den unterschiedlichsten Puzzles konfrontiert, die so noch in keinem anderen Computerspiel zu finden sind. Puzzles, die räumliches Vorstellungsvermögen, Zahlen und Wortverständnis testen sowie Rätsel in Multiple-Choice-Form. Jetzt können Home Computer-Besitzer aus erster Hand die Spannung der Fernsehserie in den eigenen vier Wänden nachleben.

ruff & Locher
datentechnik

# ST-TAST

**Die professionelle Lösung, um IBM-Tastaturen am Atari zu betreiben.**

● Keinerlei Eingriffe oder Lötarbeiten notwendig! ● Über die ST-TAST-Platine wird die Tastatur am ROM-PORT eingesteckt und mit der mitgelieferten Treibersoftware bereits beim Booten installiert. ● Sämtliche ATARI-Tasten (und Tastenkombinationen) werden unterstützt. ● Die Belegung Ihrer IBM-Tastatur kann von Ihnen geändert und Ihren individuellen Wünschen problemlos angepaßt werden.



**Neu im Programm:**
Qualitätsmarken-Tastaturen im neuesten MF-2 Design!
**CHERRY G 80-1000**
(Tastatur des Jahres '87) s. Abb.
**HONEYWELL RX-102**
(die »leise« Tastatur)

**Qualität zu Niedrigpreisen:**

| | |
|---|---|
| ST-TAST einzeln | DM 144,– |
| CHERRY G 80-1000 | DM 248,– |
| HONEYWELL RX-102 | DM 298,– |

**Paketpreis:**
**(Sie sparen DM 50,– ! !)**

| | |
|---|---|
| ST-TAST+CHERRY G 80-1000 | DM 342,– |
| ST-TAST+HONEYWELLRX-102 | DM 392,– |

Überzeugen Sie sich von der Leistungsfähigkeit und fordern Sie unser Info an!
– Händleranfragen erwünscht –

ruff & Locher
datentechnik

Eichachstraße 13
7404 Ofterdingen
Telefon
(07473) 22810



# GALACTIC

## Modulatoren für jeden Zweck

HF (Fernsehanschluß)

**MOD 1**, internes Gerät zum Einbau in 260/520/1040 — 160 DM
**MOD 2**, externes Gerät zum außen Anstecken, alle Fernsehfarben werden dargestellt, Ton kommt aus dem Fernseher — 170 DM

AV (Videoanschluß für FBAS)

**MOD 1a**, wie MOD 1, aber nur mit Videoausgang — 130 DM
**MOD 2a**, wie MOD 2, ebenfalls nur FBAS-Ausgang — 140 DM

Die Modelle MOD2/MOD 2a sind für 25 DM Aufpreis auch mit integrierter Monitorumschaltbox lieferbar!

**1 MB RAM, auch für 520 STM**
RAMs einzeln geprüft, schnelle Montage — **auf Anfrage**
Auf Anfrage auch steckbar

In Vorbereitung: 1/2/4-MB-Erweiterungen, konfigurierbar. Preis und Lieferzeit auf Anfrage!

## UNSERE PROGRAMME:

### DEEP THOUGHT 1.1

Das sensationelle Schachprogramm aus Deutschland. Neue, verbesserte Version. Spielstarker Algorithmus, frei programmierbare Eröffnungsbibliothek (bis 66.000 Positionen), echtes Blitzen möglich. Nach Aussage von Ligaspielern: Effektiv das beste Programm für Mikrocomputer! **nur 69 DM**

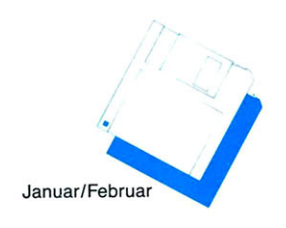
### MARS ST

Das Corewars-Programm für Ihren ST. (Siehe ST-Computer 11/87 für Details). Lassen Sie die Viren (kontrolliert) in Ihren Computer. Voller Weltmeisterschaftsstandard '86. Integrierter Editor/Assembler, sehr schnell, gutes Handbuch. Dieses Programm wurde ASM-Hit im »Aktuellen Software Markt 12/87! **nur 89 DM**

**Stachowiak, Dörnenburg und Raeker GbR**
**Burggrafenstr. 88 - 4300 Essen 1 - ☎ (0201) 27 32 90/71 0 18 30**

# ST-COMPUTER DISKETTEN-SERVICE

## DIE ST-MONATSDISKETTE



Januar/Februar

Alle in der ST-COMPUTER abgedruckten Programm-Listings (bei Compilersprachen natürlich auch die ausführbaren Programme) befinden sich auf den **ST-COMPUTER MONATSDISKETTEN.**

Auf jeder Diskette sind die Programme von zwei Monaten enthalten.

Damit entfällt das mühselige und zeitaufwendige Eintippen der Programme. **ST-MONATSDISKETTE** laden und los gehts...

Der Preis für eine **MONATSDISKETTE** (mit den Programmen von jeweils 2 ST-Ausgaben) liegt bei nur 28,— DM zuzüglich 5,— DM Versandkosten.

Es sind noch sämtliche **MONATSDISKETTEN** seit 1986 (Neuerscheinung der ST-COMPUTER Zeitschrift) erhältlich.

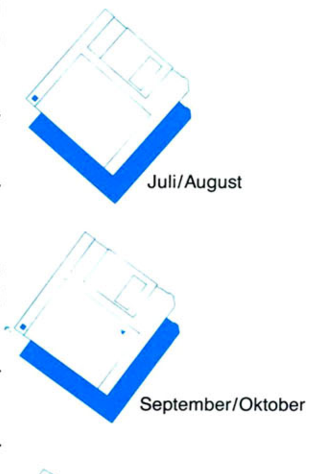
Benutzen Sie zur Bestellung die im Heft befindliche DISKETTEN-Bestellkarte.

März/April



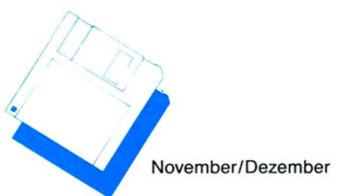
Mai/Juni



Juli/August

September/Oktober



November/Dezember

## Eine schnelle Maus

Seit einer Woche habe ich eine neue Maus an meinen ST angschlossen, die komplett funktioniert, nur einen Nachteil hat: Ihr Kugeldurchmesser ist um einiges kleiner, so daß ich fast nur mit Fingerbewegungen den Pfeil über den ganzen Bildschirm bewegen kann. Bei der Original ATARI ST Maus mußte ich wenigstens meine Hand zwischen 5 und 10 cm über den Schreibtisch bewegen.
Bei einer zügigeren Mausbewegung ( wie ich es von früher gewohnt bin ) macht nun die neue Maus nicht mehr mit, der Pfeil wandert aus oder bleibt ganz stehen.
Um dieses Problem zu beheben, muß ich Verzögerungswerte für X und Y im TOS finden und ändern. Und genau das ist der Kern meines Problems: Wo stehen diese Werte im TOS ? Nun, solltet Ihr das Problem lösen können, wäre ich Euch unendlich dankbar.

*(Robert Heinhaus, 8600 Augsburg)*

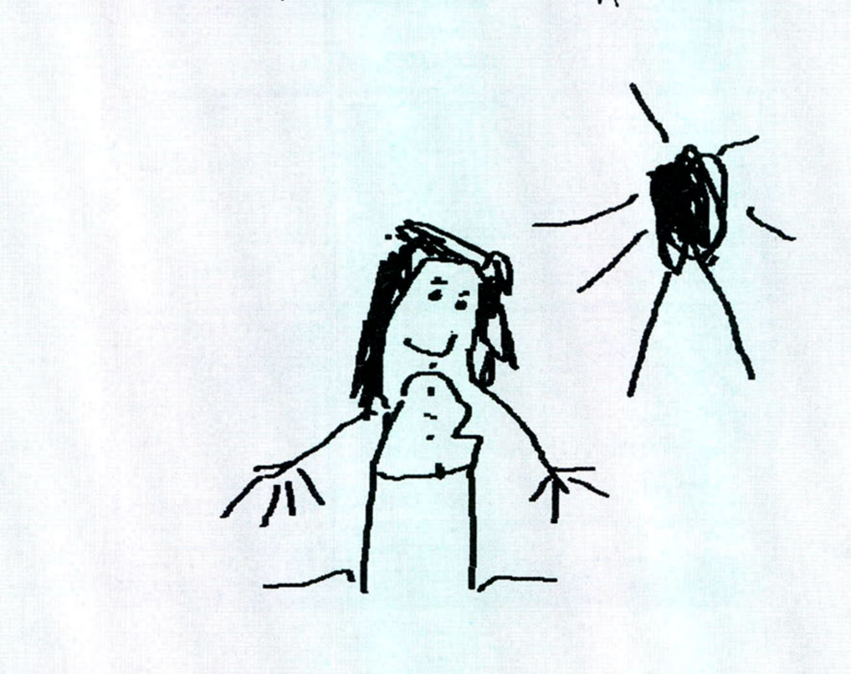
**Red.:** Leider müssen wir ein weiteres Mal darauf hinweisen, daß es nicht sehr sinnvoll ist, bestimmte absolute Adressen im Betriebssystem zu ändern, da diese mit großer Wahrscheinlichkeit in der nächsten Betriebssystemversion nicht mehr an der gleichen Stelle sein werden. Da es für solche Probleme viel elegantere Lösungen gibt und wegen des vorhergenannten Grundes, vermeiden wir es, solche Adressen bekanntzugeben.
Aber nun zu Ihrem Problem: Im Betriebssystem gibt es Routinen, die die Bearbeitung von Joystick und Maus übernehmen. Die Adressen dieser Routinen sind ganz legal über die XBIOS-Routine KBDVBASE() zu erfahren. Diese Funktion liefert einen Zeiger auf eine Tabelle, deren Inhalt die verschiedensten Zeiger, unter anderem auch einen auf die Mausroutine, enthält. Dieser Zeiger liegt 16 Bytes vom Anfang der Tabelle entfernt, ist also der fünfte Vektor. Diese Routine bekommt nun auf dem Stack - in dieser Reihenfolge - den Befehlscode mit Maustastenstatus, die relative X-Mausbewegung und die relative Y-Bewegung geliefert. Manipuliert man nun die relativen Mausbewegungen, indem man sie verdoppelt, und dann in die Originalroutine weiterspringt, so bewegt sich die Maus doppelt so schnell. Bei Ihrem Problem scheint es so zu sein, daß die Verschiebungen, die von Ihrer Maus kommen, einfach zu groß sind. Führt man eine Division der Werte durch, so müßte ihr Problem eigentlich gelöst sein. Ein ausprogrammiertes Beispiel, in der die Mausgeschwindigkeit verdoppelt wird, welches auch leicht zu ändern ist, finden Sie in der ST-Ecke der ST-COMPUTER-Ausgabe vom Februar 1988.

## Aus 20 MB mach 40 MB!

Liebe Freunde,
ein "Techniker" der Firma ATARI erklärte mir auf meine Anfrage (während der CeBit in Hannover), daß es ohne weitere Änderungen möglich sei, die 20 MB-Festplatte gegen eine 40 MB-Platte zu tauschen. Das wäre natürlich ein tolles Ding, ich kann das aber so einfach nicht glauben. Die Beantwortung dieser Frage würde sicher einen großen Leserkreis interessieren.

*(Frank Heinevetter, Celle)*

**Red.:** Einige unserer Leser dürften die Antwort auf Ihre Frage schon wissen, denn genau dieses "Problem" sind wir im Mai dieses Jahres angegangen. Im Rahmen unseres im Dezember begonnenen Festplattenkurses wurde in der ST-Computer Mai 1988 ausführlich beschrieben, wie man aus einer 20MB-Festplatte (SH204 und SH205) eine 40MB-Festplatte macht. Sicherlich wäre es untertrieben "ohne weitere Änderungen" zu sagen, aber der Aufwand hält sich tatsächlich im engen Rahmen. Es ist aber in dieser Kolumne nicht möglich, wie Sie sicherlich verstehen werden, den Vorgang zu beschreiben.



Johanna Repschläger
Blücherstr. 20
3167 Burgdorf

Burgdorf, den 01.04.1988

Liebe Leute von der ST - Computerzeitung!

Ich heiße Johanna und bin 5 Jahre alt. Gestern habe ich meinem Papa ein Bild mit der Maus auf dem Computer gemalt. Mein Papa sagt daß das so gut geworden ist, daß es Euch von der Zeitung auch gefällt. Hoffentlich mögt Ihr es auch leiden.
Das Bild heißt "Papa mit Sonne". Beim Ausdrucken und Buchstabenschreiben muß ich mir aber immer noch helfen lassen.

Viele Grüße, Eure JOHANNA

# Genauigkeit erwünscht - Die Lösung!

Letzten Monat wurden wir gefragt, wie man eine Meßgenauigkeit von einer Millisekunde in GFA-Basic erreichen kann. Obwohl wir dieses Thema in näherer Zukunft noch einmal genauer betrachten wollen, möchten wir eine von unserem Leser Rainer Kracht eingeschickte Lösung nicht unveröffentlicht lassen. Hier nun der Brief:

```
' millisec, R. Kracht, Gärtnerwiete 9, 2085 Quickborn, 29.5.88
'
Bild%=Xbios(2)              ! Adresse des Bildschirmspeichers
Zeit%=Bild%+322000          ! Adresse der Timer-A-Interrupt-Routine
Uhr%=Zeit%+20               ! Adresse der Uhr
Lpoke Zeit%, &H8B90005             ! Anfang der Interrupt-Routine
Lpoke Zeit%+4, &HFFA0F             !
Lpoke Zeit%+8, &H27CF3FF           !
Dpoke Zeit%+12, &H52B9             !
Lpoke Zeit%+14, Uhr%               ! Adresse der Uhr übergeben
Dpoke Zeit%+18, &H4E73             ! rte (Ende)
Lpoke Uhr%,0
A%=Xbios(31,0,2,246,L:Zeit%)   ! Uhr starten

Do
   Void Inp(2)
   U=Lpeek(Uhr%)                   ! Uhr lesen
   U=U+U/1024                      ! Uhr korrigieren
   U=Int(U+0.5)                    ! auf volle Millisekunden
   Print U;" msec seit Uhrenstart"
Loop
```

Ein GFA-BASIC-Programm, das eine genaue Millisekunden-Uhr in der "toten Ecke" des VIDEO-RAM installiert, könnte zum Beispiel wie im Listing aussehen.

Die DO-Loop-Schleife zeigt, wie die Uhr gelesen wird. Mit der Korrektur hat es folgende Bewandtnis. Der Quarz des MFP macht in einer Millisekunde 2457,6 Schwingungen. Der am nächsten liegende Wert, auf den sich der TIMER-A-Interrupt-Vektor einstellen läßt, ist 2460. Die damit gestellte Uhr läuft damit etwas zu langsam. Der Korrekturfaktor ist gleich 24600/24576 oder einfacher 1024/1025. Um die Rechnung weiter zu vereinfachen, läßt sich 1025/1024 auch als 1+1/1024 schreiben.

*(Rainer Kracht, 2085 Quickborn)*

Diese Routine unserer Leser (vielen Dank) müßte reichen, um sie vernünftig einsetzen zu können. Trotzdem werden wir uns in einem der nächsten Monate noch einmal ausführlich mit diesem Thema beschäftigen.

# ST-COMPUTER PUBLIC DOMAIN SERVICE

# Liebe Leser,

im letzten Monat wurden leider beim Drucken der Zeitschrift die Diskettennummern vergessen. Wir bitten dies im Nachhinein zu entschuldigen. Unsere Telefonistin hatte dadurch einiges zu tun, trotzdem waren die Probleme mit einem Anruf erledigt.
Wir haben uns deshalb entschlossen, gerade weil sich einige sehr wertvolle Programme darunter befinden, die Disketten des Vormonats noch einmal abzubilden.

Ihre ST-Computer Redaktion

## Die Disketten des Vormonats!






**SCANNER:** Das Softwarepaket zum Scannerprojekt in ST-Computer 4/88.
GEM-Oberfläche, div. Druckertreiber, Umrechnung des Bildes (Helligkeit und Kontrast) auch im Nachhinein, div. Bildgrößen, Grauwertberechnung und was sonst noch zum Scannen benötigt wird.




### Utilities

**MS-DOS STEP:** Schaltet B-Laufwerks-Step auf 6ms. Für 5.25"-Laufwerke im 40-Spur-Betrieb. Als ACC, PRG und Basic-Source. Läuft mit normalem und Blitter-TOS.




**BITTE EIN BIT 3.1:** Schnelles Kopierprogramm als Accessory. Hohe Geschwindigkeit durch Ignorieren leerer Tracks. Automatische Formaterkennung.
**FORMAT 1.0:** Formatierprogramm als Accessory. Formatiert im Hintergrund (Multitasking).
**SUPER-FILECOPY:** Filekopierprogramm. Ideal für ein Laufwerk. Kopiert erst in Puffer, dann auf neue Disk.
**MC87_1_9:** Formatier-, Kopier- und Harddiskbackupprogramm.




### Shells

**GULAM-SHELL:** Die Shell für den Programmierer (siehe ST-Computer 5/88 - Modula-Test ). Die beste Shell zum bequemen Arbeiten.
**FLESH:** Universelles Shell-Construction-Kit. Generiert Shells für alle Bereiche. Beispiel: Sie programmieren in C, Assembler und Basic; Schreiben mit 1st_Word, 2nd_Word und Signum; Zeichen mit Degas und Stad. Flesh erzeugt eine Shell, mit der Sie jedes dieser Programme per Menü anwählen können.
**TEX-SHELL:** Shell für ST-TEX, das Satzprogramm. Aufruf aller benötigten Programmteile per Menü. Gerade ST-TEX ist, bzw. war etwas kompliziert in der Bedienung.

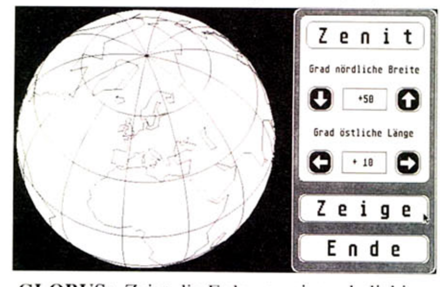



### Grafik

**SHAPES:** Das Programm läßt ein animiertes Pferd über eine Landschaft galoppieren. Sehr sehenswert. (s/w)
**MANIPULATOR:** Programm zum Verfremden von Bildern. Manipulieren aber richtig. (s/w)

### Geographieprogramme

**GEOGRAPH:** Enthält Daten von 159 Ländern. Hauptstädte, Regierungen, Produkte, Handel, Religion, Energiehaushalt, Landwirtschaft, Währung, u.a.. Abfrage der Daten in Quizform. (s/w)




**GLOBUS:** Zeigt die Erde von einem beliebigem Punkt aus dem Weltraum.




### Dateiverwaltung

**DATEI:** Flexibles Datenverwaltungsprogramm. Freie Maskenerstellung per Dialog. Import, Export von/nach anderen Dateiprogrammen, Steuerzeichenanpassung.

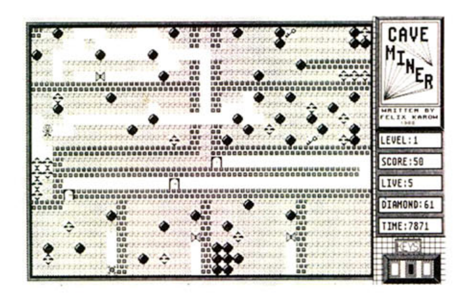



**ARTUS:** Artikelverwaltungsprogramm. Dient zum Verwalten von Artikeln aus Zeitschriften (ST-Computer und anderen). Auswahl nach Stichpunkten, Namen Monat, u.a.. Jeder Artikel läßt sich leicht finden.




### Spiele

**CAVEMINER:** Boulder Dash-Variante. Auf der Suche nach Diamanten. Doch Vorsicht vor Felssteinen, die gerne auf dem eigenen Kopf landen. Überblick, Geschick und Reaktion sind gefragt. PD-Spiel des Monats. (s/w)
**BUBBLE_EDIT:** Feld-Editor für Bubble Bobble. Schaffen Sie sich nach eigenen Wünschen ein Spielfeld mit allen Raffinessen und Gemeinheiten. (s/w)




### UPDATES

Folgende zwei Disketten wurden aktualisiert:

**141 VIRUSKILLER:** Das wichtigste Update dieses Monats ist der Viruskiller von Diskette 141. Das Programm hatte noch einen kleinen Fehler, es verdächtigte jedes kompilierte GFA-Programm einen Virus zu beinhalten. Viele Anrufer wunderten sich zurecht darüber, aber die Panik war umsonst. Leider bemerkte der Programmautor diesen Fehler erst kurz nach Erscheinen der letzten Ausgabe. Der Fehler ist inzwischen korrigiert, die Diskette seit ca. 3 Wochen aktualisiert. Die neue Version hat eine Länge von 9523 Bytes.

**125 STUDENT CAD:** Das Student CAD verträgt sich jetzt auch mit dem Blitter-TOS.

Für die Updates gelten dieselben Bedingungen, wie für die normalen PD-Disketten (siehe rechte Seite)

# Die Hundertfünfzigste

## Liebe Leser,

die 150. Public-Domain Diskette ist da. Ein guter Grund, sich einige Gedanken über die Sammlung zu machen. Vor zwei Jahren begannen wir, mit den PD-Einsendungen unserer Leser diese Sammlung zusammenzustellen. Viele Programmautoren schlossen sich an. Damals haben wir nicht geglaubt, daß eine solche Vielzahl professioneller Programme folgen würde, die dem Anwender eine Alternative zu käuflicher Software bietet. Viele Programme, gerade Utilities und spezielle Anwendungen, sind nur in der PD-Sammlung zu finden. Über die gute Qualität dieser Programme müssen wir hier wohl nichts mehr sagen. Wenn Sie einige Programme daraus kennen, werden Sie uns zustimmen. Wir hoffen, daß uns auch in diesem Jahr wieder viele Autoren helfen, die Sammlung auszudehnen. Auch wir sind gespannt, welche Programmideen bis zur 200sten Disketten verwirklicht werden.

Ihre ST-Computer Redaktion

## Neu in diesem Monat!



### Utilities

**QUICKMOUSE:** Beschleunigt die Mausgeschwindigkeit. Endlich wieder Platz auf dem Schreibtisch.

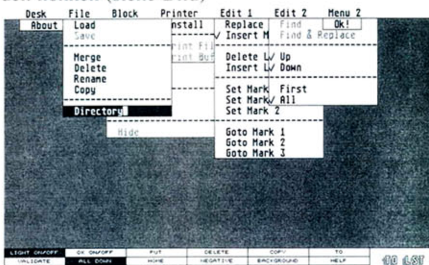
**KEY HELP:** Ermöglicht die Eingabe eines beliebigen Zeichens (ASCII-Code) nach Drücken von Alternate-Control.

DESK DATEI INDEX EXTRAS

Hexdump von F:\GFABASIC.PRG

Dateilänge: 56409 Bytes Infos mit <Help>

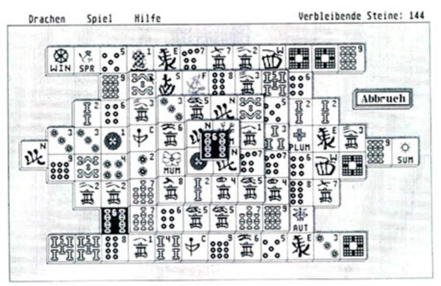
**DUMP:** Accessory, mit dem eine beliebige Datei angezeigt und einzelne Daten darin geändert werden können (siehe Bild)




**MENÜEDIT:** Eine Art RCS für GFA-Basic-Menüleisten. Gestalten Sie Ihre Menüleiste per Dialog. Das Programm erzeugt sodann einen kompletten Programmkopf für GFA-Basic, der in das eigene Programm eingebunden wird und die gesamte Menüführung Ihres Programms erledigt. Lohnenswert. (s/w)

**TEMPELMON:** Verfeinerte Version des besten ST-Monitors/Debuggers. Kommentar überflüssig, man kommt nicht daran vorbei.

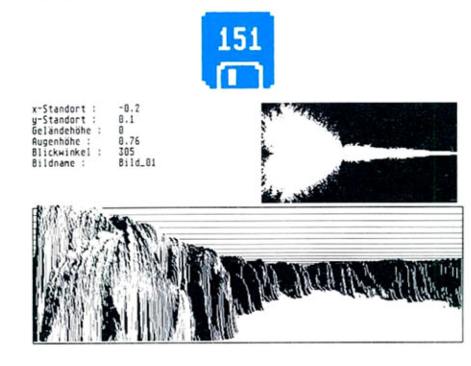


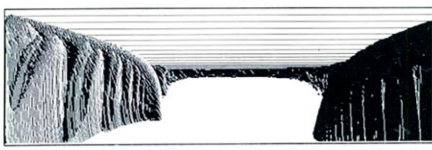



### Spiele

**DRACHEN:** Umsetzung eines 2000 Jahre alten Brettspiels. Durch Suchen zweier freiliegender Randsteine muß der Steineberg abgebaut werden. Doch das ist nicht immer so einfach. Grafisch sehr schön gestaltet. (s/w)

**SHANG:** Ähnlich Drachen, unterschiedlicher Bildaufbau. Welches der beiden Programme nun besser ist, konnten wir nicht entscheiden. Dies überlassen wir Ihnen. (s/w)



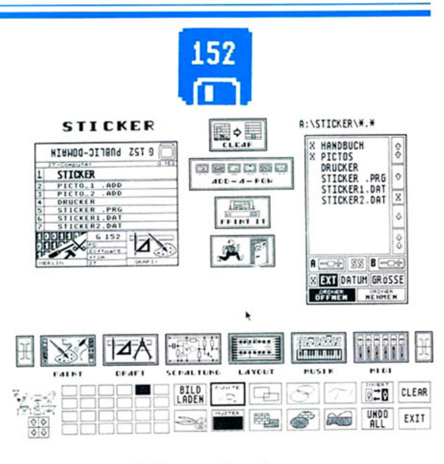



### Fractals

**3D-FRACTALS:** Programm erzeugt ein dreidimensionales Fractalbild. Der Blickwinkel kann im Grundbild in jeder Richtung und Höhe gewählt werden. Sehr interessantes Programm. Begeben Sie sich in die Welt der fraktalen Berge. (s/w)

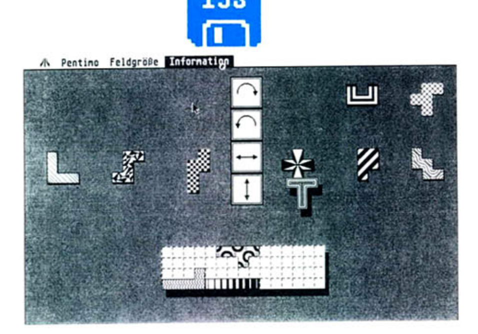
## Machen Sie mit

Möchten Sie ein selbstgeschriebenes Programm in die PD-Sammlung geben? Kein Problem. Schicken Sie es uns auf einer Diskette zu, samt einer Bestätigung, daß es von Ihnen erstellt wurde und frei von Rechten Dritter ist.




### Etikettendrucker

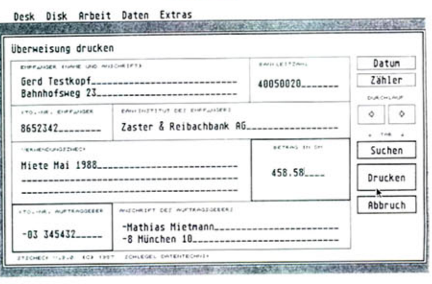
**STICKER:** Disketikettendrucker mit sehr vielen Optionen. Liest Directory ein und übernimmt per Mausklick Programmnamen in die Etiketten. Jedes Etikett mit Bild. Bilder zu vielen Gebieten vorhanden. Mit integriertem Zeichenprogramm, zum Zeichnen der Grafiken, die auf das Etikett gedruckt werden (siehe Bild). (s/w)




### Puzzlespiel

**PENTIMO:** Verschiedene Puzzleteile müssen in einem beliebig großen Feld untergebracht werden. Teile dreh- und rotierbar. Feldgröße veränderbar. Die Lösung ist nicht immer einfach. Benötigt daher viel Hirnzeit. Sehr schön gestaltet, ideal für langweilige Sommerabende. (s/w)






### Anwendersoftware

**SCHECK:** Druckprogramm für Schecks und Überweisungsaufträge. Leicht bedienbar. Anpassung an jedes Formular durch Verschiebung der Datenpositionen per Maus. Speicherbare Stammdaten (Kontonummer, Bankleitzahl, Namen, etc.).

**DRUCKER:** Druckeranpassungsprogramm für SCHECK. Anpassung an jeden Drucker. Viele Anpassungen schon vorhanden( z.B. NL10, P6,..)

Zeichenerklärung:

s/w = nur monochrom
f = nur Farbe

# ST-COMPUTER PUBLIC DOMAIN ALTBEWÄHRTES

## Der Service der Redaktion

**Die nicht aufgeführten Disketten sind natürlich auch weiterhin erhältlich, nur wurde uns der Platz zu klein. Schauen sie dazu bitte in einer der vorherigen Ausgaben nach oder fordern Sie die Liste an (frank. Rückumschlag).**
**Wir haben Ihnen hier einige der besten PD-Disketten aufgeführt, die trotz Alter bislang unübertroffen sind.**

**Diskette 7**
- **XLISP 1.71:** (die Sprache der künstlichen Intelligenz) incl. 54 KByte Dokumentation, Anleitung und Beispielen.

**Diskette 11**
- **PROLOG 10:** Die Sprache der 5. Generation. Unterstützt GEM. incl. 95 KByte Anleitung und Beispiele.

**Diskette 23**
- **Das große deutsche Ballerspiel:** besonderes Aktionspiel (f)

**Diskette 34**
- **Assemblerpaket:** nach M 68000-Standard (2-Pass Assembler, Disassembler, Debugger, Editor)

**Diskette 40**
- **Deluxe Fontmaster ST:** Druckprogramm für Karten oder schönen Tabellen. Viele Zeichensätze, Proportionalschrift und Blocksatz, eigener Zeichensatzeditor.

**Diskette 75**
- **Randvoll mit Zeichensätzen** für Fontmaster (Disk 40).

**Diskette 49, 50, 51**
- **Volksforth 83:** Die drei Disketten beinhalten ein komplettes, leistungsstarkes Forth-System für den ST. Alle Diskette werden benötigt.

**Diskette 52**
- **Molekül:** Darstellung chemischer Moleküle. In s/w und Farbe wobei in Farbe mit 3D-Effekt. Atom, Ionen oder Van der Waalsradius. Ein Molekül kann gedreht und sogar animiert werden.

**Diskette 61**
- **Hacomini:** Erzeugt Hardcopies in Miniaturformat (48mm * 68mm) (s/w).
- **Helper:** Diskmonitor und Druckerutility.
- **Bidipo:** Scrollt bis über 100 Bilder (4MB) über den Schirm. Für Diashows.
- Datobert : Grafische Darstellung. Linien- und Kuchengrafik, sowie in dreidimensionaler Balken- und Flächengrafik. (s/w)
- **Direct all:** Directorydruck.
- Überweisung: Überweisungsdruckprogramm.

**Diskette 70**
- **Orbit:** Satellitenbahnberechnung. Animation über der Weltkarte (s/w)
- **High End 2.2:** Programm zum Berechnen von Lautsprecherboxen aller Art. Lautsprecher und Boxenmaße. Datei für Speaker enthalten. (s/w)

**Diskette 77**
Utilitydisk
- **FCOPY 2.0:** Schnelles Kopierprogramm. Formatieren, Verify, Multiple Copy, Scan und Teilkopie.
- **Speeder:** Verdoppelt Schreib- und Lesegeschwindigkeit.
- **Filecopy:** Kopieren von Einzeldateien. Auch mit einer Floppy.
- **Formater:** Erhöht die Diskkapazität
- **Fastformat:** Formatiert Schnelladedisks.

**Diskette 81**
Malprogramm
- **Public Painter:** Superzeichenprogramm. Gepacktes Format, SNAPSHOT-Bilder, Lasso, Blockmanipulation, Vergrößern, Verkleinern, Drehen, Biegen, Verzerren, Überschneiden. Drucken von Bildausschnitten.(s/w)

**Diskette 86**
- **CARPET:** Berechnungsprogramm von 3D-Funktionen. Hidden-Line Algorithmus und Beleuchtungseffekten. Funktionseingabe und frei einstellbare Parameter. (s/w)
- **3D-PLOT:** Ähnliches Programm in compiliertem GFA-Basic, mit einigen Extra-Funktionen. (s/w)
- **Funktionsplot:** Funktionseingabe im Programm, verschiebbare Achsen, mehrere Darstellungsarten (s/w)

**Diskette 88**
Druckertreiber
- **NEC-EMU:** Hardcopytreiber für NEC P5/P6/P7. Erzeugt Hardcopies in 12 verschiedenen Größen und versch. Kontrasten. Enthält Druckerspooler.
- **LQ 800:** Hardcopytreiber für EPSON 24 Nadeldrucker in drei verschiedene Größen.
- **NEC_CODE:** Ein Programm zur Einstellung des NEC P6 /P7 u.ä..
- **SIMPLE:** Komfortables Hardcopy-Programm mit vielen Extras.
- **STX-80:** Hardcopytreiber für diesen Thermodrucker.
- **WORDPLUSTREIBER:** Druckertreibersammlung für folgende Drucker: NL10, FX1000, SP1000I.

**Diskette 90**
Utilities
- **MAXIDISK 4:** Die komprimierende Maxidisk mit max. 4MByte Kapazität. Läuft auf ALLEN TOS-Versionen. Komprimiert automatisch ihren Inhalt. RESETFEST.
- **TOPS:** The Other Pascal Shell (für PascalPlus), TEMPUSfähig, erzeugt Crossreferenz. Kopieren, Löschen und Umbenennen von Files.
- **SIGNUMSHELL:** Eine Shell zu Signum. Umschaltung zwischen Signum Editor, den verschiedenen Druckprogrammen, den Zeicheneditoren per Menüleiste. Automatisches Install.
- **TEMPELMON 1.8:** DER weitverbreitete Monitor.
- **RAMDISK:** Gleiches Programm wie MAXIDISK, nur ohne Komprimieralgorithmus.
- **COPY:** Beide Ramdisk enthalten ein Autocopyprogramm zum kopieren bestimmter Files in Ramdisk.
- **RCS.WANDEL:** Wandelt *.h Files nach Basic und Assembler.

**Diskette 93**
- **ADR_2:** Semiprofessionelles Adreßverwaltungsprogramm mit Serienbrieferstellung (mit und ohne 1st Word). Edierbare Druckeranpassung (Steuerzeichen, Zeichenwandlung), erzeugt Adreß-Telefon und Geburtstagslisten. Druckt Labels, Etiketten nach freidefinierbarem Format. (s/w)
- **DAT_TEXT:** Adressverwaltung für Mehrpersonenzugriff auf eine Datei. Datenzugriff nur durch Passwort (s/w)

**Diskette 100**
- **ST-SPEECH:** Sprachausgabe auf Softwarebasis. Wandelt englischsprachige Texte automatisch in verständliche Lautschrift. (s/w)
- **FORTRAN SHELL:** Eine Shell für Fortran 77. Bequemes Aufrufen von Compiler, Linker, und anderen Programmen.
- **LIFE IS LIFE:** Life-Simultionsprogramm. Sechs verschiedene Feldgrößen, Drehen, Verschieben, Kopieren von Blöcken wie im Malprogramm, Festlegen der Fortpflanzungsregeln, hohe Geschwindigkeit. (s/w)
- **FASTLIFE:** Life-Simulation in Volksforth. Wahnsinnige Geschwindigkeit. (s/w)
- **GEMFRAC:** Berechnung fraktaler Landschaften in Farbe und s/w.
- **SHARP:** Basic Programme für Sharp-Basicrechner. Geeignet für Direktüberspielung mit Interface.

**Diskette 113**
- **ZEITMANAGER:** Semiprofessionelle Terminplanverwaltung. Eigener Desktop, Terminerinnerung, Listendruck, Timingverwaltung, fixe Termine, Kalender. Schaltjahrfest (s/w)

**Diskette 116**
- **µEMACS 3.8:** Deutsche Version. Eigene Macrosprache, Textverschlüsselung, Mailmerge, mehrere Screens, u.v.a.m.

**Diskette 117**
Terminalprogramm
- **UNITERM:** Exclusives Terminalprogramm. VT200, VT102, VT100, VT52, 4010, u.a. Softscroll, Grafikübertragung, Funktionstastenbelegung,..

**Diskette 120**
- **WISSEN SIE ES?:** Quizspiel mit vielen Fragen aus verschiedenen Gebieten. Ähnlich 'Trivial Pursuit'. Viele Fragen aus verschiedenen wissensgebieten, eigene integrierbar. (s/w)

**Diskette 121**
Spiele
- **GO_UP:** Loderunner-ähnliches Spiel. Auf vielen Mauern, Leitern und Seilen müssen Sie den Verfolgern entkommen. Viele Level. Editor zum Entwickeln eigener Levels enthalten. (s/w)
- **MAZE_EDIT:** Editor zu dem Spiel 'MIDI MAZE'.
- **F1 MANAGER:** Verwaltung eines Formel-Eins-Rennstall. (s/w)
- **OELIMPERIUM:** Managen einer Ölfirma. (s/w)

**Diskette 122**
Spiel
- **PD_BOLO:** Sie kennen Arkanoid, dann sollten Sie auch PD_Bolo kennen. Interessante Break-Out Variante.

**Diskette 123**
Utilities
- **Hyperformat** Formatierprogramm von Claus Brod.
- **FSELECT** Die neue Fileselectorbox. Wird fest installiert. Kommt bei jedem Aufruf. Wer sich noch mit der alten herumquält ist selbst dran schuld.
- **Diskmon** Diskettenmonitor. (s/w)
- **PC_Ass** Assembler für Sharp-Taschencomputer.

**Diskette 125**
Zeichenprogramm
- **CAD 2:** Objektorientiertes Grafikprogramm, z.B. mit Zoom ohne Auflösungsverlust.(s/w)

**Diskette 126**
- **CUBE HACK:** Terminalprogramm mit integrierter, umfangreicher Programmiersprache. Anwendungen: z.B. Automatisches Einloggen in Mailboxen.
- **OR Spread:** Spreadsheetähnliches Programm zur Linearoptinmierung, Netzplanentwicklung, Kürzeste Wege Berechnung. (s/w)

**Diskette 127**
Wissenschaftliche Anwendungen
- **Planet:** Himmelskörperberechnungen. (Grafische Darstellung des Sternenhimmels, Horizont). Animation.
- **Turing:** Realisierung eines Turing-Maschinen-Modells (s/w)
- **Zust:** Analyse und Simulation linearer Regelkreise (s/w)

**Diskette 128**
Spielesammlung
- **Horrorschloß:** Der Weg durch viele Gänge verbirgt viele Gefahren.
- **Inversi:** Das bekannte Spiel, gegen den Rechner oder zu zweit (s/w)
- **Invation:** Space Invadors mit 3-D Vektorgrafik. (s/w)
- **Zarge:** Ein 32000K-Bild wird gepuzzelt (s/w)

**Diskette 129**
Utilities
- **GFA-Shell:** Shell für GFA-Basic, incl. CLI und Sprite-Editor (s/w)
- **OMIKRON-Shell:** Shell für OMIKRON.-Basic System. Aufruf von Interpreter, Compiler, Linker. Generiert PRG, TOS, TTP oder ACC. (s/w)
- **Disksave:** Speichert Bildschirm per 'Alt+Help' auf Disk.
- **Apple-ST:** Datenübertragung Apple II—> ATARI ST.

**Diskette 130**
Utilities
- **HFORMAT:** Formatierprogramm als Accessory.
- **Rescue:** Rettet gelöschte Files. (s/w)
- **Diskinfo:** Gibt Directory sortiert auf Drucker (Liste oder Label) bzw. Datei aus. (s/w)
- **AccLoader:** Ermöglich die Auswahl bei mehr als 6 Accessories (s/w)

**Diskette 131**
**Macros zu VIP-Professional**
(professionelle Beispiele zum VIP-Kurs im Heft) (1MByte Ram)
- **Verein:** vollständige Vereinsverwaltung. Serienbriefe, Lastschrifteinzug, Überweisungen, Etikettendruck, Listendruck.
- **Meister:** komplette Verwaltung einer Vereinsmeisterschaft.

**Diskette 132**
Simulationen
- **Wator:** Simulation eines biologischen Systems, am Zusammenleben von Haien und Fischen.
- **Life:** Eine sehr flexible Life-Version. (s/w)
- **Quicklife:** Rasend schnelles Life. (s/w)

**Diskette 133**
- **Codewars:** REDCODE-Programm. Der Kampf im Rechner. Incl. ausführlich Einführung in die Theorie.
- **Kerne 2.1:** Erweiterte Version der 'Krieg der Kerne'.

**Diskette 134**
Spielesammlung
- **Galerie:** Wunderschöne Patience-Version. Für kalte Frühlings-Abende.
- **Kreuz As:** Umsetzung des Spielautomaten. Mut zum Risiko ist angesagt. (s/w)

**Diskette 135**
- **Midi-Musik:** Komfortables Musik-Programm. Eingebauter Notendruck (s/w)

**Diskette 136**
- **IQ-Test:** Berechnet Ihren IQ. Was da wohl rauskommt? (s/w)
- **GENIUS:** Leistungsstandes, Reaktionstests, Gedächnisschulung, Auswertung der Ergebnisse. (s/w)
- **DIFFENENZIAL:** Ableitung, totales und partielles Differenzial. Ausgabe als Funktionsgleichung. (s/w)
- **KLIMA:** Grafische Ausgabe von Klimatabellen der ganzen Welt. (s/w)
- **ZAHN:** Stirnrad-Vorauslegungsberechnung. Hoffnung für Maschinenbaustudenten. (s/w)

**Diskette 137**
Druckertreiber
- **STAR LC10:** Verschiedene WORDPLUS-Treiber für LC10.
- **STAR NL10:** Mehrere WORDPLUS-Treiber für NL10.
- **CITIZEN:** WORDPLUS-Treiber für 120D, LSP 10 und LSP 11.
- **GABIGRAF:** Grafiktreiber für Typenradmaschine und Matrixdrucker. Gibt Grafik und vielseitig gestaltbaren Text aus.

**Diskette 138**
- **MEMORY:** MEMORY-Brettspiel. Mehrere Kartenpakete enthalten. Eigene Spiele leicht erstellbar. Feldgröße und Anzahl der Karten frei wählbar. Für Kinder und Gedächnisgenies. (s/w)

**Diskette 139**
- **DGDB** Das Große Deutsche Ballerspiel in monochromer Version (Farbversion auf PD23). Aktionspiel. Für 1 und 2 Personen (s/w)
- **SCHLOSS** Textadventure mit Sprachausgabe. Abenteuer im unheimlichen Schloß des großen Zauberers.

**Diskette 140**
Haushaltsverwaltung
- **ETAT:** Planung und Kontrolle des eigenen Haushalts. Buchung, Monatsplanung, Kontenverwaltung, Jahresplanung und Jahresabrechnung.
Bringt die Haus-Kasse unter Kontrolle. (s/w)

**Diskette 141**
Utilities
- **COMMAND:** Commando-Interpreter mit vielen Funktionen: Packen/entpacken, verschlüsseln, Funktionstastenbelegung, Find, Passwortabfrage, History-Funktionen, Zahlenumwandlung, Batchprozessing (Loop, Goto, Vergleichsoperationen, Variablen)
- **MONOMON:** Monitor für Floppy, Festplatte und Ramdisk. (s/w)
- **ANTIVIRUS:** Einmal geladen, meldet es alle Viren im Bootsektor. Boot with Antivirus...
- **NOBACKUP:** Löscht *.BAK-Dateien
- **HYP-COPY:** Kopierprogramm für Hyperformat-Disketten.
- **S-FORMAT:** Formatieren einzelner oder mehrerer Sektoren. (s/w)
- **TOTALDELETE:** Physikalisches Löschen von Files.

**Zeichenerklärung**

*(s/w) - Nur Monochrom*
*(f) - Nur Farbe*
*kein Kürzel - Farbe und Monochrom*

## Sonderdisks

Die folgenden Programme sind nicht Public-Domain.
Sie können aber bei uns bezogen werden.
**A.) TOS:** Die letzte Disketten-Version vom 6.2.1986. Ältere Versionen laufen nicht problemlos.
Unkostenbeitrag DM 15.-
**B.) RCS:** Das Resource-Construction Set aus dem ATARI-Entwicklungspakets. Unverzichtbar bei der GEM-Programmierung.
Unkostenbeitrag DM 15.-
Bis auf die Höhe des Unkostenbeitrages gelten die gleichen Versandbedingungen wie bei der Public-Domain Software.

**VERSANDBEDINGUNGEN**

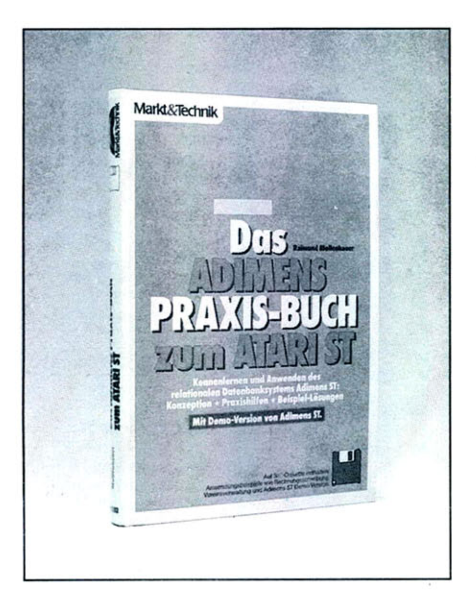
Sämtliche Disketten können ab dem Erstverkaufstag der ST-Computer direkt bei der Redaktion bezogen werden. Wir haben für Sie den schnellstmöglichen Versandservice eingerichtet. Lieferung innerhalb einer Woche.

**1. Schriftliche Bestellung**
- Der Unkostenbeitrag für eine Diskette beträgt DM.-.
- Bezahlung nur per Scheck oder Nachnahme
- Im Ausland nur Vorrauskasse möglich)
- zuzüglich DM 5,00 Versandkosten (Ausland DM 10.-)
- bei Nachnahme zuzüglich DM 3,70 Nachnahmegebühr
- Legen Sie bitte, falls zur Hand, einen Aufkleber mit Ihrer Adresse bei.
- Bitte fügen Sie keine anderen Bestellungen oder Anfragen bei.

**2. Anruf genügt**
'MERLIN'-Computer GmbH
ST-Computer Redaktion
Tel.: 0 61 96 / 48 18 11
Von Mo-Fr 9.00 bis 17.00 Uhr

Die Bezugsadresse lautet:
'MERLIN'-Computer GmbH
ST-Computer Redaktion
'PD-Service'
Postfach 5969
D-6236 Eschborn

Bei Fragen bezüglich der Programme stehen wir Ihnen gerne zur




*Raimund Mollenhauer*
*Das Adimens Praxis-Buch*
*zum ATARI ST*
*Haar b. München,*
*Markt & Technik Verlag 1988*
*ISBN 3-89090-552-8*

Mit diesem Werk ist das erste Buch zur relationalen Datenbank Adimens ST erschienen. Zum Autor wäre zu sagen, der Name Raimund Mollenhauer dürfte einigen Leuten schon etwas sagen, denn er war von 1985 - 1987 bei ATARI Deutschland für die Anwendungssoftware zuständig. Dort leitete u.a. Schulungen für Adimens, daß ja von ATARI vertrieben wird. Seit Januar 1988 ist er nun bei ADI-Software, dem Hersteller von Adimens, für Datenbanksysteme zuständig.

Am Anfang des Buches wird zuerst ein Überblick über die Anwendungsmöglichkeiten von Adimens gegeben. Anschließend folgt etwas Theorie über verschiedene Datenbank-Modelle und wie man überhaupt eine Datenbank für den eigenen Gebrauch entwerfen sollte. Dies ist vor allem für den Anfänger wichtig, damit er nicht planlos drauflos konstruiert. Dies geschieht sinnvollerweise anhand des Init-Moduls von Adimens.

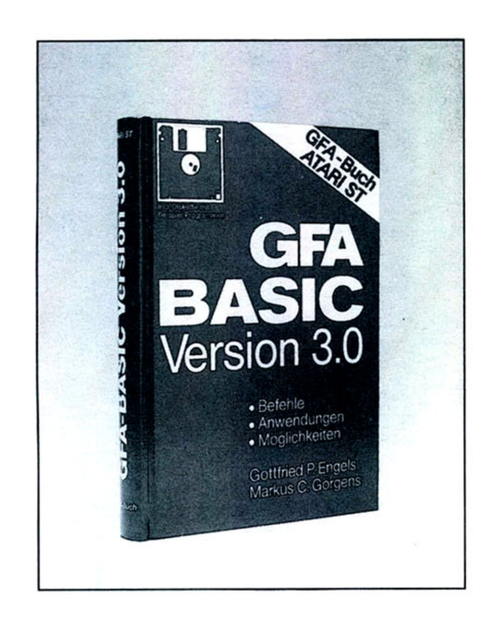
Als nächstes werden die Datenbankfunktionen von Adimens ausführlich erklärt. Das reicht vom normalen Eingeben, Suchen, usw. bis zur Listenerstellung, Berechnung und Import/Export von Daten.

Anschließend werden in den Kapiteln 3 und 4 Anwendungsbeispiele für Rechnungsschreibung, Verkaufsübersicht und Vereinsverwaltung gegeben. Dabei wird auch erklärt wie man die Masken und Felder einer bereits bestehenden Datenbank nachträglich verändern kann, falls man halt doch etwas bei der Planung vergessen hat.

Um zu einem Fazit zu kommen: das Buch kann jedem weiterempfohlen werden, der schnell eine Einführung in Adimens bekommen möchte und dem es zuviel Arbeit ist, das ganze Handbuch durchzuarbeiten.

*HE*

---




*Gottfried P. Engels / Markus C. Görgens*
*GFA-BASIC Version 3.0*
*Düsseldorf,*
*GFA Systemtechnik GmbH 1988*
*ISBN 3-89317-004-9*

Selten wird von einem Softwarehersteller ein Produkt auf den Markt gebracht und gleichzeitig dazu ein passendes Buch von ihm angeboten. Noch seltsamer ist, daß dieses Softwareprodukt dann auch noch mit einem sehr umfangreichen Handbuch geliefert wird. Aber genau das ist im Falle des GFA-BASIC-Buchs eingetreten, das wir Ihnen vorstellen möchten.

Denn gleichzeitig mit der Lieferung der neue Version des GFA-BASIC (V3.0) wird von GFA Systemtechnik ein Buch angeboten, das die Bedienung des mittlerweile sehr umfangreichen GFA-BASIC-Interpreters Version 3.0 erläutert.

Das Buch ist kein Ersatz zu dem Original-Handbuch sondern eher als eine Ergänzung zu ihm zu verstehen. Deswegen werden logischerweise nicht die einzelnen Befehle des Interpreters bis ins Detail erklärt. Womit beschäftigt sich dann dieses Buch? Prinzipiell mit den neuen Eigenschaften dieser Version sowie mit Programmiermethodik.

So wird zum Beispiel ein ganzes Kapitel dem Thema Programmstrukturierung gewidmet. Hier wird noch einmal die Bedeutung der 'Strukturierung' betont, aber es wird auch gezeigt, wie man strukturiert programmiert. Nicht nur die Strukturbefehle, die es in früheren GFA-BASIC-Versionen gab, sind hier besprochen, sondern auch die, die neu hinzugekommen sind. Ebenso verhält es sich bei dem Kapitel Integerarithmetik.

Fast alle Kapitel werden mit einem etwas längeren und sinnvollen Beispiel abgeschlossen. So wird die Sache nicht zu trocken und die Augen haben etwas zu sehen.

Für wen ist dieses Buch gedacht? Im Prinzip für jeden, der diese Version des GFA-BASICs gekauft hat. Der Anfänger wird mit Sicherheit zuerst in das Handbuch sehen, und wenn das nicht mehr ausreicht, hat er mit diesem Buch ein weiteres Werk, das ihm noch zusätzliche Hilfe leisten wird. Der Insider wird in diesem Buch viele Anregungen für eigene Projekte finden. Der absolute Kenner,.. na ja, wenn es so was gibt....

*MM*

# Immer up to date

Mit dieser Sparte wollen wir allen unseren Lesern die Möglichkeit geben, sich über die neuesten Programm-Versionen zu informieren. Angegeben werden die aktuelle Versionsnummer, ein eventueller Kopierschutz, die Bildschirmauflösungen und der Speicherbedarf. Softwarefirmen ist es somit möglich, die ST-Computer-Leser über ihre Updates zu informieren.

| Programmname | Version | Daten | Programmname | Version | Daten |
|---|---|---|---|---|---|
| Adress ST / Check ST | 1.0 | N H | Link_it GFA | 1.1 | N HML |
| AnsiTerm | 1.1 | N | Link_it Omikron | 2.0 | N HML |
| 1st BASIC Tool | 1.1 | N HML | Lisp Complete | 1.01 | N HM |
| Binlook | 1.0 | N HML | Lock_it | 1.0 | J HML |
| BTX für ST | 1.0 | N H 1M | Megamax C | 1.1 | N HML |
| Crypt_it | 1.0 | J HML | Mega Paint | 1.0 | N H 1M |
| Flash-Cache/Flash-Bak | 1.0 | N HM | Megamax Modula 2 | 3.5 | N HM |
| Flexdisk | 1.2 | N HML | Micro C-Shell | 2.70 | N HM |
| 1st_Freezer | 1.0 | N HML 1M | MT C-Shell | 1.10 | N HM 1M |
| GFA-Artist | 1.0 | N L | Multi ST | 1.0 | N HML 1M |
| GFA-BASIC 68881 | 1.3 | N HML | Musix32 | 1.01 | J H |
| GFA-Compiler | 2.02 | N HML | Omikron BASIC Compiler | 2.0 | N HML |
| GFA-Draft | 2.1 | N | Omikron BASIC Interpreter | 2.0b | N HML |
| GFA-Draft plus | 1.5 | N | PAM's TERM/4014 | 3.012 | N H |
| GFA-Interpreter | 2.02 | N HML | PAM's TurboDisk | 1.7 | N HML |
| GFA-Objekt | 1.1 | N HM | PAM's NET | 1.0 | N HML |
| GFA-Starter | 1.1 | N HML | PC ditto Euroversion | 3.64 | N HML |
| GFA-Vektor | 1.0 | N | Pro Sound Designer | 1.2 | L |
| Hard Disk Accelerator | 1.0 | N HML | Pro Sprite Designer | 1.0 | L |
| Hard Disk Toolkit | 1.04 | N HM | Search! | 1.31 | N HM |
| Harddisk Utility | 2.0 | N HM | Signum! zwei | 1.0 | N H |
| Imagic | 1.0 | N HML | 1st_Speeder | 1.01 | N HML |
| Intelligent Spooler | 1.01 | N HML | 1st_Speeder 2 | 1.0 | N HML 1M |
| K-Resource | 1.1 | N HM | STAD | 1.3 | N H |
| Label ST | 1.0 | N HML | ST Pascal plus | 2.02 | N HM |
| Laser Deluxe | 1.0 | N HML 1M | Tempus | 2.0 | N HM |
| 1st_Lektor | 1.2 | HM | Transfile ST plus | 2.0 | N HM |
| Lern ST | 1.21 | N HML | VSH Manager | 1.11 | N HML 1M |

Irrtum vorbehalten
Daten-Legende: N = kein Kopierschutz, J = Kopierschutz, H = hohe Auflösung, M = mittlere Auflösung, L = niedrige Auflösung, 1M = mindestens 1 Megabyte Speicher notwendig

# INSERENTEN-VERZEICHNIS

# In der nächsten ST-Computer lesen Sie unter anderem

## Sound-Sampler im Eigenbau

Eine der interessantesten und vielseitigsten Einsatzmöglichkeiten von Computersystemen ist die Erfassung und Verarbeitung von analogen Meßdaten. Zu diesen zählt man digitale Filter, Speicher-Oszilloskope und Sound-Sampling. Aber auch moderne CD-Player verdanken ihre hohe Qualitätsstufe der Digitaltechnik.
Allen diesen Anwendungsmöglichkeiten liegt das gleiche Prinzip zugrunde; die Umsetzung von elektrischen Spannungen als "Rohmaterial" in digitale Daten mittels mathematischer Berechnungen. Wir möchten Ihnen in der nächsten Ausgabe die nötige Hard- und Software zum Bau eines eigenen 8-Bit-Sound Samplers vorstellen.

## DFÜ im Griff - Telekommunikationsprogramm Interlink ST

Interlink ST ist ein nicht mehr ganz neues, aber sehr interessantes Telekommunikationsprogramm mit einem Leistungsumfang, der fast keinen Wunsch mehr offen läßt. Interlink ist nicht nur ein Terminalemulator; doch dazu mehr in der nächsten Ausgabe.

## Aditalk-Kurs

Zweifellos hat das Datenbanksystem Adimens eine breite Akzeptanz bei den ATARI-Anwendern gefunden. Einen wesentlichen Einfluß hat dabei die integrierte Kommandosprache Aditalk, die über die Befehle zur Datenmanipulation hinausgeht und eine komplette Anwendungsprogrammierung zuläßt.
Aus diesem Grund beginnen wir in der nächsten Ausgabe einen Kurs über flexible Modulprogrammierung mit Aditalk, der den Anfänger in Adimens einführen und die Kenntnisse beim Fortgeschrittenen vertiefen soll.

Änderungen vorbehalten !

## Die ST-Computer Ausgabe 8/9 '88 erscheint am 29.7.1988

## Fragen an die Redaktion

Ein Magazin wie die ST-Computer zu erstellen kostet sehr viel Zeit und Mühe. Da wir weiterhin bemüht sind, die Qualität zu steigern (auch wenn das manchmal daneben geht), haben wir Redakteure ein großes Anliegen an Sie, unsere Leser:

Bitte Haben Sie dafür Verständniss, daß Fragen an die Redaktion in Zukunft nur noch Donnerstags von $14^{00}$- $17^{00}$ Uhr telefonisch beantortet werden können.

*Vielen Dank für Ihr Verständnis*

*Ihre Redaktion*


## Impressum ST Computer

**Chefredakteur:**
Uwe Bärtels (UB)

**Redaktion:**
Uwe Bärtels (UB)
Harald Egel (HE)
Marcelo Merino (MM)
Harald Schneider (HS)

**Redaktionelle Mitarbeiter:**
C.Borgmeier (CBO)
Claus Brod (CB)
Stefan Höhn (SH)
Raymund Hofmann (RH)
Oliver Joppich (OJO)
Jürgen Leonhard (JL)
Claus P. Lippert (CPL)
Markus Nerding (MN)
Chr. Schormann (CS)
Andreas Suchy (AS)
Jörg Wilhelm (JW)

**Autoren dieser Ausgabe:**
H.H.Ackermann
D.Brockhaus
I.Brümmer
M.Erprath
A.Esser
P.Fischer
W.Keller
M.Kofler
R.Köhling
D.Rabich
H.J.Reiss
M.Schuhmacher (MS)
H.D.Schultz (HDS)
M.Spehr
K.Steger
R.Tolksdorf
V.Ullrich

**Public Relations:**
Claus P. Lippert (Leitung)
D.dela Fuente (UK)
L.Hennelly (Nordamerika)

**Redaktion: "Merlin" Computer GmbH**
Postfach 59 69
Industriestr. 26
6236 Eschborn
Tel.: 0 61 96/48 18 11
FAX : 0 61 96/4 11 37

**Verlag:**
Heim Fachverlag
Heidelberger Landstr. 194
6100 Darmstadt 13
Tel.: 0 61 51/5 60 57
FAX : 0 61 51/5 56 89 + 5 60 59

**Verlagsleitung:**
H.J.Heim

**Anzeigenverkaufsleitung:**
U.Heim

**Anzeigenverkauf:**
K.Magaritis

**Anzeigenpreise:**
nach Preisliste Nr.3, gültig ab 1.1.88
ISSN 0932-0385

**Grafische Gestaltung:**
Fabian & Mayer

**Fotografie:**
R.Spirandelli, Archiv

**Produktion:**
K.H.Hoffmann, B.Failer, S.Failer

**Druck:**
Ferling Druck

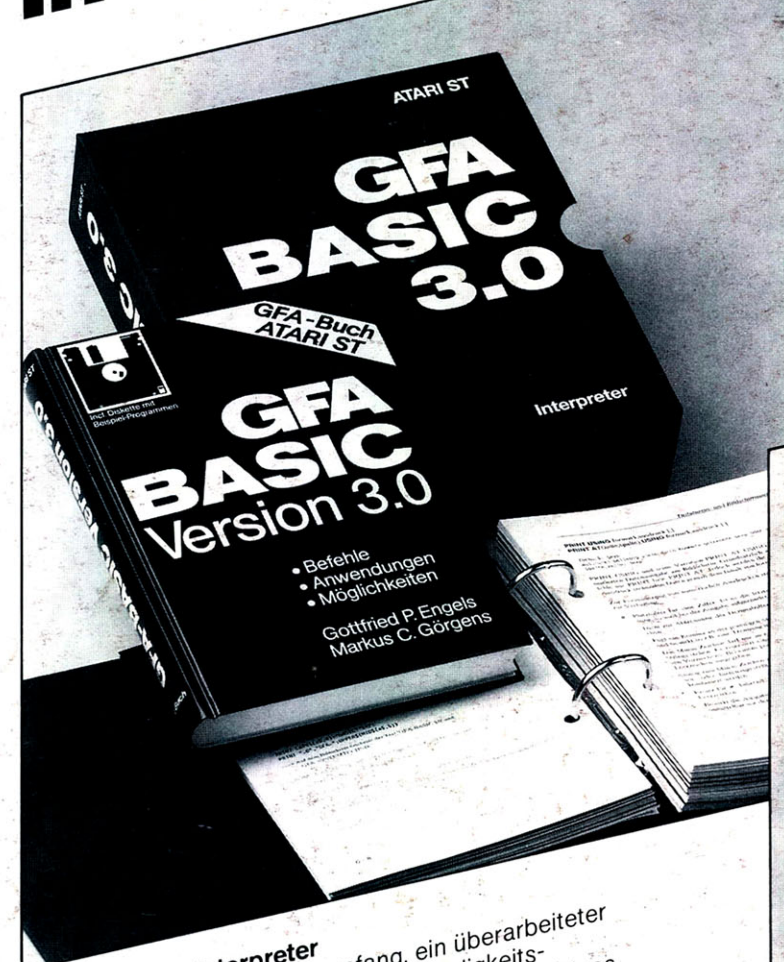
**Lektorat:**
Vera Pfeiffer

**Bezugsmöglichkeiten:**
ATARI-Fachhandel, Zeitschriftenhandel, Kauf- und Warenhäuser oder direkt beim Verlag

**ST Computer erscheint 11 x im Jahr**
Einzelpreis: DM 7,-, ÖS 56,-, SFr 7,-
Jahresabonnement: DM 70,-
Europ. Ausland : DM 90,-
Luftpost : DM 120,-

**Manuskripteinsendungen:**
Programmlistings, Bauanleitungen und Manuskripte werden von der Redaktion gerne angenommen. Sie müssen frei von Rechten Dritter sein. Mit seiner Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung auf Datenträgern dem Heim Verlag.
Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.

# Programmieren in GFA-BASIC




**GFA-BASIC 3.0 Interpreter**
Ein stark erweiterter Befehlsumfang, ein überarbeiteter Editor, eine erneute, deutliche Geschwindigkeitssteigerung und ein komplett neu erstelltes, 550-seitiges Handbuch: Das sind die herausragenden Merkmale der neuen Version 3.0 von GFA-BASIC. DM 198.-

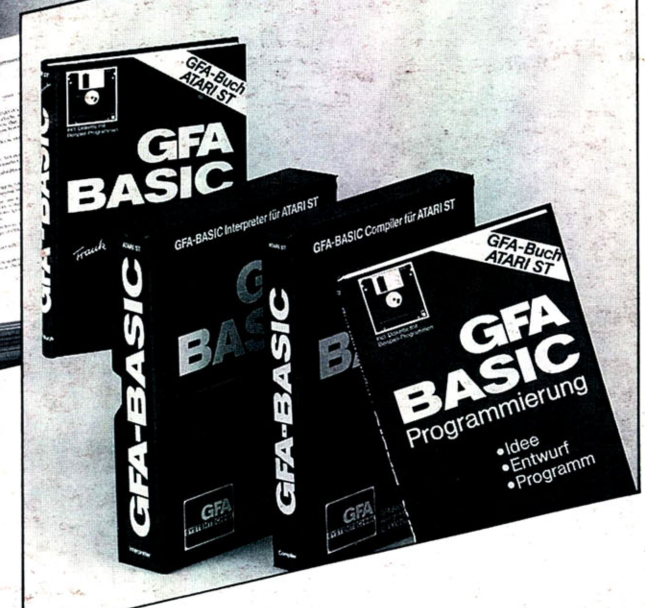
**GFA-BASIC-Buch: Version 3.0**
Für alle, die bisher mit GFA-BASIC gearbeitet haben und nun den optimalen Umstieg auf die neue Version 3.0 suchen, haben die Handbuch-Autoren Gottfried P. Engels und Markus C. Görgens dieses Buch geschrieben. Anhand vieler Beispiele werden die Möglichkeiten der neuen Anweisungen von GFA-BASIC 3.0 gut verständlich vermittelt. DM 59.-
ISBN 3-89317-004-9
400 Seiten, inkl. Diskette

**GFA-BASIC 2.0 Interpreter**
Das meistverkaufte Programm für ATARI-ST zum Einsteigerpreis DM 99.-

**GFA-BASIC 2.0 Compiler**
Mit diesem Compiler werden Ihre GFA-BASIC 2.0-Programme noch schneller und sind selbständig lauffähig. DM 99.-

**GFA-BASIC-Buch**
Frank Ostrowski über sein GFA-BASIC. Sie erhalten viele wertvolle Tips vom Autor des GFA-BASIC. Besonders geeignet für den fortgeschrittenen Anwender.
ISBN 3-89317-001-4 DM 79.-
288 Seiten, inkl. Diskette

**GFA-BASIC-Programmierung**
Von der Idee über den Entwurf bis hin zum fertigen Programm. Sie lernen mit Hilfe dieses Buches das strukturierte Programmieren in GFA-BASIC.
ISBN 3-89317-003-0 DM 49.-
288 Seiten, inkl. Diskette




## GFA-CLUB
## GFA-ST/PC-Software
## bitte Info anfordern

**...Anruf genügt.**

*GFA Systemtechnik GmbH*
*Heerdter Sandberg 30*
*D-4000 Düsseldorf 11*
*Telefon 02 11/58 80 11*

